TUGAS AKHIR
(NA 1701)

PERHITUNGAN HIDROSTATIK UNTUK BADAN KAPAL YANG DIBENTUK DARI KURVA B-Spline

Oleh :

DARWIN.L.RUNTU
NRP 4195 100 504

JURUSAN TEKNIK PERKAPALAN
FAKULTAS TEKNOLOGI KELAUTAN
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
S U R A B A Y A

1999

JURUSAN TEKNIK PERKAPALAN

FAKULTAS TEKNOLOGI KELAUTAN ITS

SURAT KEPUTUSAN TUGAS AKHIR (NA 1701)

No. : 135 /PT12.FTK2/M/1997

Nama Mahasiswa	:	Darwin Leopold Runtu
Nomor Pokok	:	4195100504
Tanggal diberikan tugas	:	01. Nopember. 1997
Tanggal selesai tugas	:	15. Pebruari. 1998
Dosen Pembimbing	:	1. Ir. Tri Achmadi, Ph.D
		2. Ir. P. Eko Pamunggal, Ph.D

Uraian / judul tugas akhir yang diberikan :

RHITUNGAN HIDROSTATIK UNTUK BADAN KAPAL YANG DIBENTUK DARI KURVA B- Spline#

sOn

Surabaya, 04 N o p e m b e r 1997
Jurusan Teknik Perkapalan FTK-ITS
K e t u a,

Ir. Koestowo Sastro Wiyono.
NIP. 130 687 430

Tembusan :
1. Yth. Dekan FTK-ITS.
2. Yth. Dosen Pembimbing.
3. Arsip.

TUGAS AKHIR
(NA 1701)

PERHITUNGAN HIDROSTATIK UNTUK BADAN KAPAL YANG DIBENTUK DARI KURVA B-Spline

Disetujui oleh :

18/9 99

Ir. TRI ACHMADI Ph.D
NIP. 130 286 963

19/9 99

Ir. P.EKO PANUNGGAL Ph.D
NIP. 131 782 033

KATA PENGANTAR

Puji syukur pada saat ini , penulis panjatkan kepada Tuhan Yang Maha Esa, yang telah memberikan kemampuan untuk meyelesaikan Tugas Akhir ini. Dimana hanya karena rahmat dan bijaksana dan semata-mata atas kehendakNya Tugas Akhir ini dapat dirampungkan.

Tuga Akhir ini dengan judul "**PERHITUNGAN HIDROSTATIK BADAN KAPAL YANG DIBENTUK DARI KURVA B-Spline**"ini disusun sebagai salah satu syarat untuk menyelesaikan studi di Fakultas Teknologi Kelautan Jurusan Teknik Perkapalan Institut Teknologi Sepuluh Nopember Surabaya, sebagai prasyarat kesarjanaan.

Selanjutnya penulis meyampaikan rasa terima kasih kepada :

- ≈ Bapak, ibu, kakak-kakak dan adikku yang telah memberikan dukungan moral maupun material sehingga penulis dapat menyelesaikan studi dengan lancar.
- ≈ Bapak **P.Eko .Panunggal ,PhD**, selaku Dosen Pembimbing I Tugas Akhir.
- ≈ Bapak **T.Achmadi ,PhD**, selaku Dosen Pembimbing II Tugas Akhir.
- ≈ Bapak **Ir. Soejitno** selaku Dosen Wali.
- ≈ Bapak **Ir. Koestowo Sastro Wiyono** , selaku Ketua Jurusan Teknik Perkapalan, FTK, ITS.
- ≈ Seluruh warga WHISPERZ I/99, Bendul Merisi II/2
- ≈ Rekan P-32, khususnya bung **Jerry Silalahi** dan P-34 kang **Eko.sJ**.
- ≈ Semua pihak yang tidak sempat tersebutkan satu-persatu yang telah membantu penulis.

Akhirnya penulis hanya bisa berharap semoga tulisan inio bermanfaat bagi pembaca. Dan dengan segala kerendahan hati penulis menerima segala kritik dan saran untuk perbaikan Tugas Akhir ini.

Surabaya, alfa Augstus 1999

Penulis

ABSTRAK

Hidrostatik merupakan salah satu elemen yang tidak dapat tidak diperlukan untuk dipakai untuk dasar perencanaan kapal.

Dengan dimulai memberi input yaitu berupa ½ lebar badan kapal yang kemudian diolah menjadi persamaan parametris dan diproses dengan surface fitting sehingga menghasilkan vertek dalam ruang 3-D, dari data tersebut kemudian diolah melalui sejumlah rumusan, baik itu berupa luas, volume dan momen inersia yang pada akhirnya menghasilkan ke-18 item kurva hidrostatik.

Program ini diujikan dengan ½ bola dan perbedaanya dinyatakan dalam standart deviasi yang bervariasi mulai dari 6 digit dibelakang koma, hingga 1 digit dibelakang koma. Dimulai dari yang terkecil KB hingga WSA.

Dari pengujian dan perbandingan program ini jika dibandingkan dengan tabulasi terhadap perhitungan analitis, didapatkan ketelitian yang bervariasi yaitu dimulai dari yang paling terkecil standart deviasinya yaitu KB sebesar 0.00000198 kemudian yang menengah yaitu TBM, TKM, LBM, LKM , Cb, Cp, Cm, Cw, berkisar antara 0.0002 s/d 0.0006, dan dilanjutkan yang paling besar standart deviasinya yaitu WPA, Displacemen dan WSA yaitu berkisar antara 0.03 s/d 0.7.

DAFTAR ISI

BAB I.
PENDAHULUAN

I.1. Latar Belakang Masalah .

Berbicara tentang kurva kurva hidrostatik maka tentu yang bayangan yang menyertai yaitu sekumpulan kurva kurva , yang dimana kurva kurva tersebut menunjukkan karakteristik karakteristik kapal dalam berbagai sarat air.

Pembuatan kurva kurva hidrostatik yang lazim dilakukan seperti pada aktivitas mahasiswa Perkapalan khususnya semester ± III ,dengan menggunakan teknik integrasi. Teknik integrasi yang digunakan adalah dengan memakai methode Simpson, tepatnya dalil Simpsons yang ke I.

Methode yang sama juga dipakai oleh penulis disini, untuk menghasilkan kurva kurva hidrostatik tetapi yang membedakan adalah pembagian baik terhadap panjang kapal dan lebar kapal ,dimana sebelumnya badan kapal yang ada dibagi menjadi ± 24 station dan ± 10 garis air (WL) , maka pada kesempatan ini penulis membagi kapal menjadi 191 station dan 191 garis air . Maksud pembagian yang lebih rapat ini adalah disengaja, dimana diharapkan dengan hal tersebut didapatkan hasil yang lebih akurat , serta jumlah station dan garis air yang ganjil bukan genap adalah merujuk kepada dalil Simpsons I.

Berangkat dari pembagian yang lebih rapat tersebut , maka dilakukan pengolahan data ½ lebar (3-D) kapal yang dihasilkan oleh persamaan permukaan B-Spline , dan dengan methode integrasi Simpsons serta ditambah sedikit perhitungan aritmatik maka akan dihasilkan ke –18 item kurva kurva hidrostatik.

I.2. Batasan Masalah

Pembatasaan disini meliputi :

- ½ lebar yang digunakan adalah hasil dari hasil persamaan permukaan B-Spline
- methode integrasi yang dipakai adalah dalil Simpsons pertama.
- Koefisien blok yang dipakai yaitu 0.6 dan 0.65
- Perhitungan sepenuhnya didasarkan dari ½ lebar kapal yang dipakai sebagai input program kurva kurva hidrostatik ini .

I.3. Tujuan

Penulis mempunyai tujuan agar hasil yaitu kurva kurva hidrostatik akan lebih akurat dan tentunya juga presisi, dimana hal ini terkait pada kemampuan internal komputer.
Hasil dari program kurva kurva hidrostatik ini berupa data , dimana dari data tersebut dapat ditampilkan dilayar ataupun untuk dicetak pada lembaran.

I..4 Manfaat.

Dengan telah terbentuknya ke-18 item kurva kurva hidrostatik tersebut maka dapat dilakukan analisa lebih lanjut, misalkan untuk studi mengenai stabilitas sebuah kapal ataupun kelak dari beberapa program-program yang terkait dengan analisa kapal dapat dibuat menjadi sebuah paket program, misalkan SFOLDS (Naval Architecture Design Analysis System) , NASTRAN ataupun yang lainnya, dimana dalam paket program

tersebut sudah terdapat lines plan ,kurva kurva hidrostatik , diagram stabilitas hingga perhitungan kekuatan memanjang kapal yang tentunya akan sangat membantu dalam menganalisa keberadaan kapal dengan segala misterinya.

BAB II
REPRESENTASI PERMUKAAN DENGAN B-SPLINE

II.1. Teori Dasar.

Suatu permukaan dapat dibentuk dengan methode B-Spline berdasar cara yang sama dengan pembentukkan kurva, hanya saja untuk suatu permukaan digunakan dua buah parameter yakni u dan w. Dengan dua parameter ini maka akan terbentuk dua fungsi basis dan dua knot vektor. Permukaan B-Spline dibentuk dari persamaan:

$$Q(u,w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j}\ N_{i,k}(u)\ M_{j,l}(w)$$

dimana Ni,k(u) dan Mj,l (w) masing masing adalah fungsi Basis B-Spline untuk parameter u dan arah w, dengan persamaan yang sama dengan basis untuk kurva, yaitu :

$$N_{i,1}(u) = \begin{cases} 1, \text{ jika } x_i \le u \le x_{i+1} \\ 0, \text{ selainnya} \end{cases}$$

$$N_{i,k}(u) = \frac{(u - x_i)\, N_{i,k-1}(u)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - t)\, N_{i-1,\,k-1}(u)}{x_{i+k} - x_{i+1}}$$

dan

$$M_{j,1}(u) = \begin{cases} 1, \text{ jika } x_i \le w \le y_{j+1} \\ 0, \text{ selainnya} \end{cases}$$

$$M_{j,l}(w) = \frac{(w - y_j)\, M_{j,l-1}(w)}{y_{j+l-1} - y_j} + \frac{(y_{j+l} - t)\, M_{j-1,\,l-1}(w)}{y_{j+l} - y_{j+1}}$$

dimana xi dan yi masing- masing adalah elemen elemen knot vektor [X] dan [Y], masing-masing untuk parameter u dan w . Sedang Bi,j adalah titik- titik dari jaring jaring poligon pembentuk permukaan. Harga n+1 dan m+1 adalah jumlah titik poligon pada arah parameter u dan w. Sebagaimana pada kurva B-Spline bentuk dan karakter permukaan B-Spline ditentukan oleh knot vektor yang digunakan, dimana dapat dipilih knot vektor periodik, open atau nonuniform . Tipe knot vektor yang dipakai tidak harus sama untuk arah parameter u dan w , meski belum umum diambil tipe yang sama untuk kedua arah parameter tersebut. Sifat yang dimiliki permukaan B-Spline tidak jauh berbeda dengan sifat-sifat B-Spline diantaranya:

- Harga order terbesar yang dapat diambil untuk masing-masing arah parameter adalah sama dengan jumlah titik poligon pembentuk pada arah tersebut.
- Kontinuitas permukaan pada masing- masing arah parameter adalah order minus satu.
- Pengaruh dari suatu titik jaring-jaring poligon terbatas pada ± ½ dan ± ½ dari bentangan pada masing- masing arah parameter.

Turunan parametris dari permukaan B-Spline dapat diperioleh sebagai berikut:

$$Qu(u,w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} Bi,j \; Ni,k'(u) \; Mj,l(w)$$

$$Qw(u,w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} Bi,j \; Ni,k(u) \; Mj,l'(w)$$

$$Quw(u,w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} Bi,j \; Ni,k(u) \; Mj,l(w)$$

{turunan- turunan dari fungsi basis sebagaimana pada persamaan}

II.2 B-Spline Surface Fitting

Sebagaimana pada kurva, jaring- jaring poligon pembentuk permukaan B-Spline juga dapat dibentuk jika seperangkat data titik dari permukaan yang akan dibentuk diberikan, dimana hasil permukaan bentukkan merupakan interpolasi dari data-data titik yang diberikan . Harga fungsi- fungsi basis Ni,k (u) dan Mj,l (w) dapat dihitung dari order yang diambil , jumlah titik poligon pada tiap-tiap titik data yang diberikan. Dari persamaan .....dapat ditulis :

$D_{1,1}(u_1,w_1) =$

$N_{i,k}(u_1)\,[M_{1,1}(w_1)\,B_{1,1} + M_{2,1}(w_1)\,B_{1,2} + \ldots + M_{m+1,1}(w_1)\,B_{1,m}] +$
.....

.....
$N_{n+1,k}(u_1)\,[M_{1,1}(w_1)\,B_{1,1} + M_{2,1}(w_1)\,B_{1,2} + \ldots + M_{m+1}(w_1)\,B_{n+1,m+1}]$

Dimana untuk r X s titik data maka harus dipenuhi $2 \le k \le n+1 \le r$ dan $2 \le k \le m+1 \le s$. Persamaan diatas diteruskan hingga sejumlah titik data yang diberikan. Dalam bentuk matrik dapat dituliskan :

$[D] = [C]\,[B]$

dimana $Ci,j = Ni,k \,.\, Mj,l$

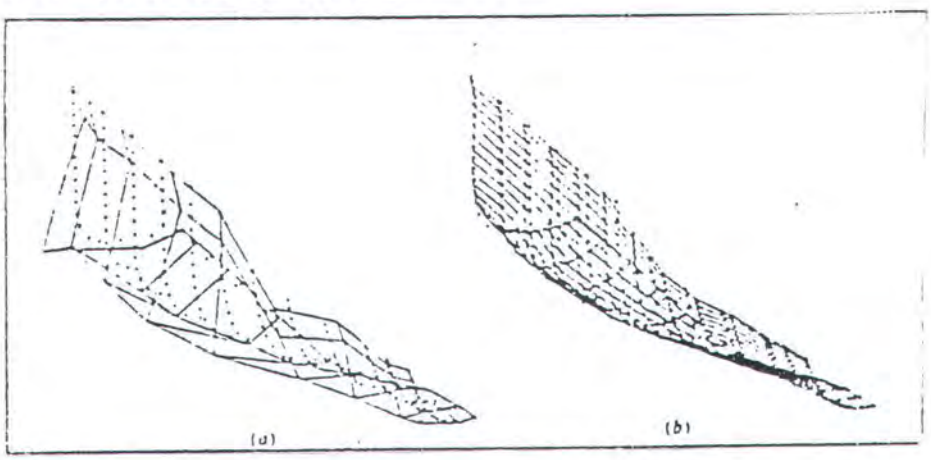
untuk r x s titik data , matrik [D] berorde (r x s) x 3 yang berisi koordinat 3 dimensi dari titik permukaan. Matrik [C] akan berorde (r x s) x (n x m), jika matrik [B] berorde (n x m) x 3 yang berisi koordinat dari titik- titik jaring- jaring poligon yang dicari.

Sebagaimana pada curve fitting, jika [C] bujursangkar yaitu r x s = n x m , maka [B] dapat langsung diperoleh dengan invers matrik .

$$[B] = [C]^{-1} [D]$$

dan jika [C] tidak bujursangkar maka penyelesaiannya:

$$[B] = [\ [C]^T\ [C]\ ]^{-1}\ [C]^T\ [D]$$



Gb.2.1. Permukaan B-Spline dengan teknik fitting
a. titik data dan poligon yang diperoleh
b. titik data dan hasil permukaan yang terbentuk

harga parameter u dan w untuk tiap-tiap titik data dapat diperoleh menggunakan suatu pendekatan panjang chord antar titik data. Dimana untuk sejumlah titik data pada arah u harga parameter u pada titik data ke 1 adalah :

$$u_1 = 0 \ ; \ \frac{u_1}{u_{max}} = \frac{\sum_{g=2}^{l} D_{g,s}\ D_{g-1,s}}{\sum_{g=2}^{r} D_{g,s}\ D_{g-1,s}}$$

dengan cara yang sama , untuk s titik data pada arah parameter w :

$$w_1 = 0 \; ; \; \frac{w_1}{w_{max}} = \frac{\sum_{g=2}^{l} D_{r,g} \; D_{r,g-1}}{\sum_{g=2}^{r} D_{r,g} \; D_{r,g-1}}$$

dimana u_{max} dan w_{max} adalah harga maksimum pada knot vektor bersesuaian.

II.3. Pembentukkan potongan badan kapal dari persamaan B-Spline

Dengan telah terbentuknya persamaan dari badan kapal , dimana setiap titik pada permukaan telah tercakup dalam persamaan tersebut , maka berbagai potongan dari badan kapal berupa body line, water line, maupun buttock line pada sebarang tempat dapat diperoleh. Hal ini akan memberikan kemudahan pada tahap produksi kapal, antara lain bentuk gading pada section tertentu bisa diperoleh dengan cepat.

Untuk memperoleh potongan pada harga x, y, dan z tertentu dari persamaan B-Spline dari badan kapal tidak bisa secara langsung diperoleh karena persamaan tersebut adalah persamaan parametris dengan variabel bebasnya adalah u dan w, bukan x,y,z. Sedangkan antara u dan w tidak terdapat suatu fungsi atau rumus yang menghubungkan, dimana u dan w menunjukkan panjang lengkungan kurva pada masing arah dan xyz menunjukkan panjang proyeksi kurva tersebut pada sumbu xyz. Teknik yang bisa digunakan disini

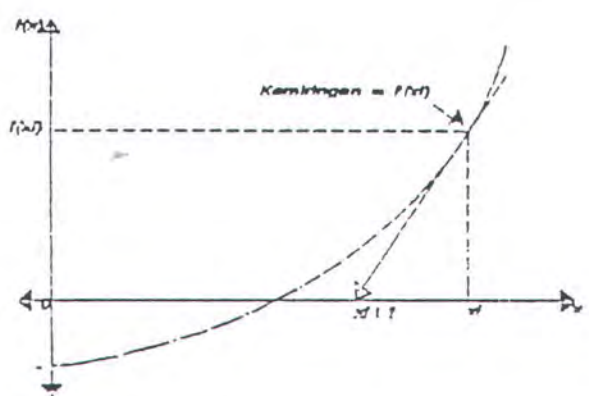
dalah untuk mencari potongan pada harga x atau y atau z tertentu adalah melakukan erasi dari harga u dan w.

'eknik iterasi yang digunakan disini adalah teknik iterasi untuk memperoleh akar-akar ersamaan dari suatu kurva. Hanya saja jika untuk mencari akar-akar persamaan dari suatu urva pada harga nol dari salah satu variabel, maka sini yang dicari adalah pada harga ertentu dari salah satu variabel. Beberapa methode iterasi antara lain Methode bagi dua, Aethode Posisi Palsu dan Metode Newton – Raphson.

Dalam pembahasan ini yang digunakan adalah metode yang terakhir yaitu Newton Raphson dimana hanya diperlukan satu taksiran awal.

II.4.Metode Newton – Raphson.

Dari terkaan awal pada akar xi,dapat ditarik suatu garis singgung dari titik (xi(f(xi)). Titik potong dari garis singgung ini dengan sumbu x biasanya merupakan taksiran akar yang lebih baik.



Gb.2.2.. Prinsip metode Newton – Raphson.

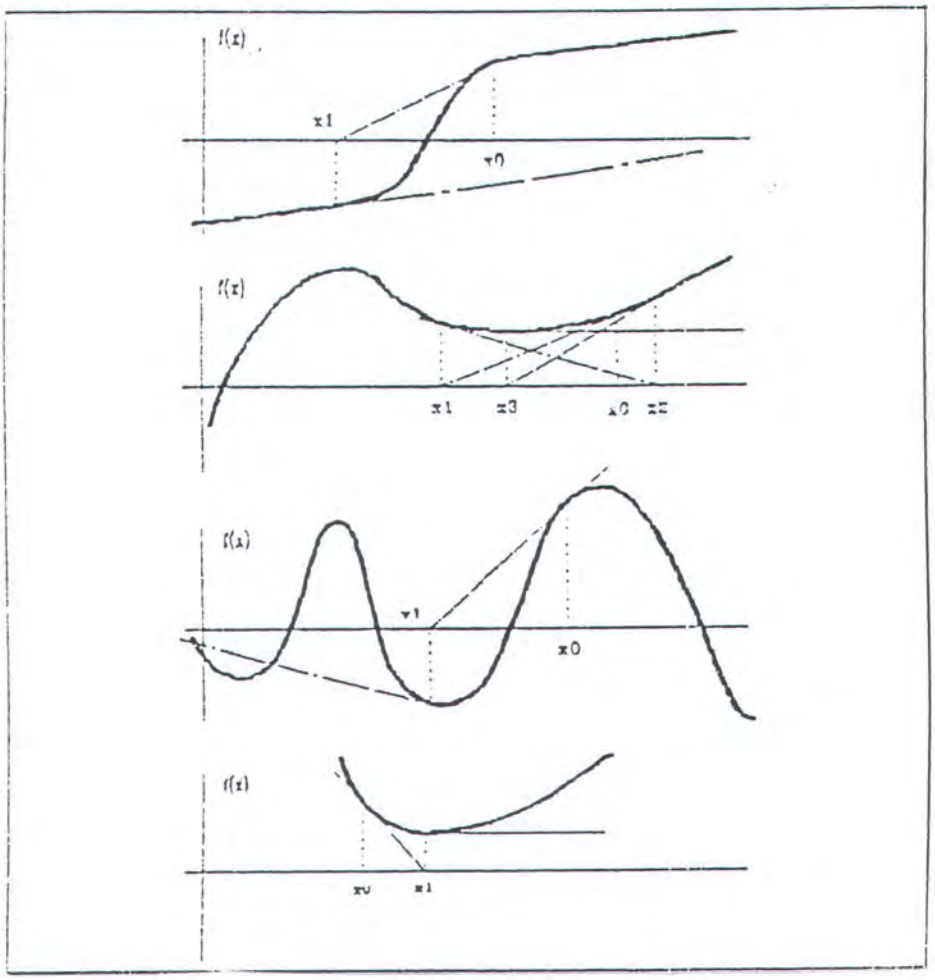
Tangen dari garis singgung tersebut adalah turunan dari fungsi kurva yang dapat dituliskan:

$$F(xi) = \frac{f(xi) - 0}{xi - xi+1}$$

Kemudian dapat disusun sebagai :

$$Xi+1 = xi - \frac{f(xi)}{f'(xi)}$$

persamaan diatas adalah rumus dari metode Newton – Raphson ini biasanya sangat efisien, hanya saja terdapat keadaan dimana metode ini akan berjalan dengan buruk pada bentuk bentuk kurva tertentu. Beberapa contoh bentuk kurva yang dapat menjebak metode ini dapat dilihat pada gambar berikut:



Gb2.3.. Kasus kasus yang menjebak pada penggunaan metode Newton – Raphson

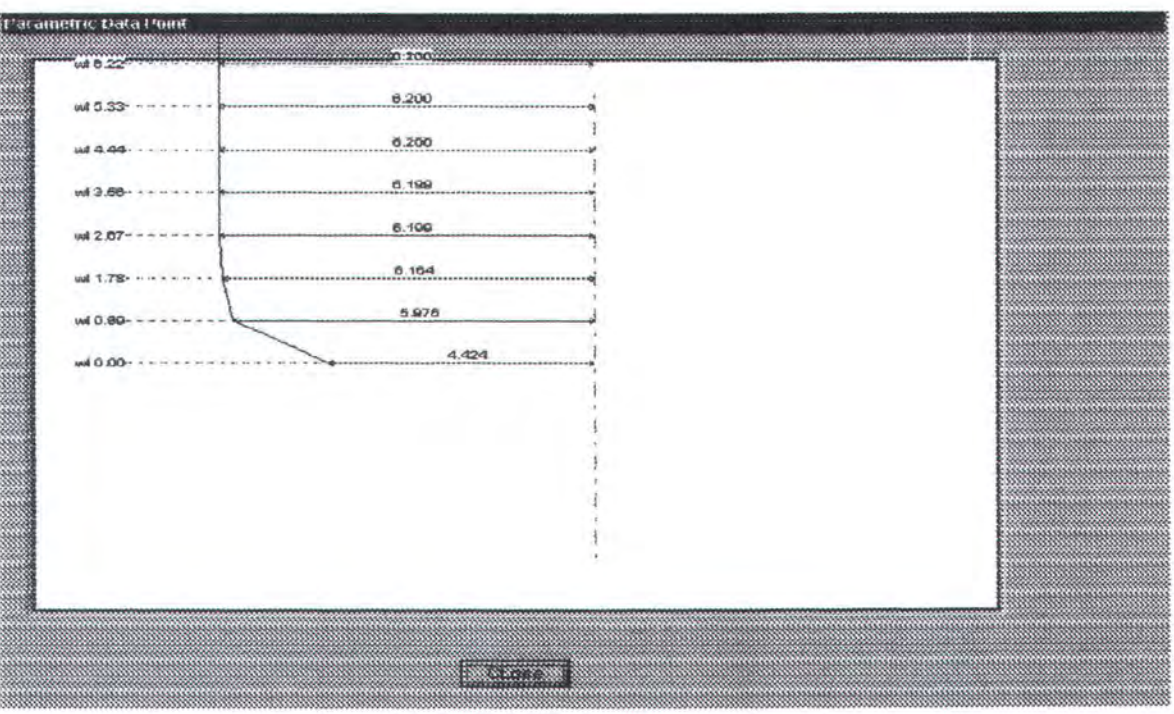
Pembentukkan potongan melintang dan mendatar .

Potongan melintang dan mendatar pada badan kapal akan membentuk kurva body line dan kurva water line. Kedua kurva ini sama-sama digunakan untuk menunjukkan harga y atau setengah lebar pada harga x dan z tertentu. Pada body line potongan diambil pada harga x tertentu dan kemudian dibentuk kurvanya dari harga y pada harga harga z tertentu. Sedangkan untuk water line potongan diambil pada harga z tertentu dan kurvanya terbentuk dari harga y pada harga-harga x tertentu. Dengan demikian untuk pembentukkan

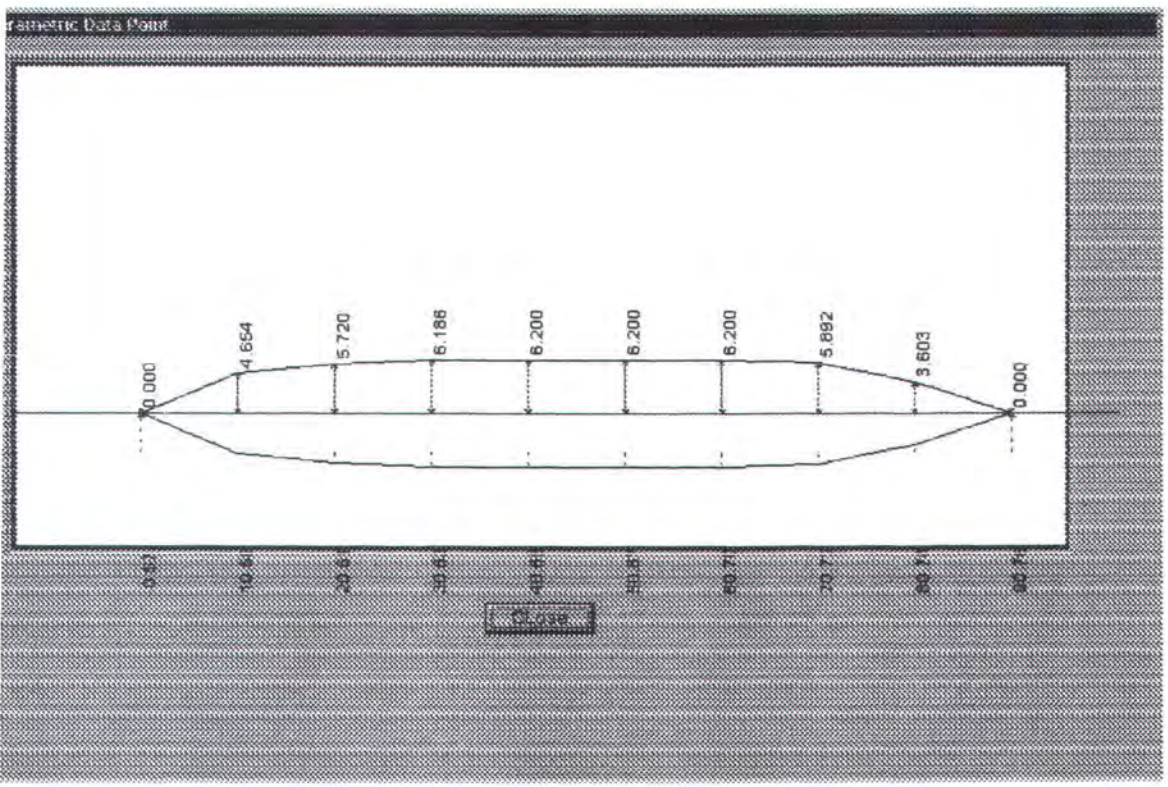
kedua kurva ini dapat digunakan suatu prosedur yang sama , yaitu suatu prosedur untuk mencari harga y pada harga x dan z yang ditentukan.

Untuk mencari harga y tersebut maka dilakukan iterasi terhadap parameter u untuk mendapatkan harga x yang ditentukan dan secara serentak juga dilakukan iterasi terhadap parameter w untuk mendapatkan harga z. Dari langkah itu diperoleh parameter u dan w yang menunjukkan letak titik pada permukaan pada harga x dan z yang ditentukan. Dari harga parameter u dan w tersebut bisa diperoleh beberapa harga y yang bersesuaian.

Parameter u dihubungkan dengan variabel x karena kedua variabel ini mempunyai hubungan yang unik , dimana proyeksi dari lengkungan pada arah parameter u tertentu terhadap bidang y=0 hampir membentuk garis yang sejajar dengan sumbu koordinat x. Sedang antara parameter w dan variabel z juga mempunyai hubungan yang hampir sama, hanya ada kemungkinan terjadi jebakan. Namun jebakan ini dapat diatasi dengan sebelum melakukan iterasi.



Gb.2.4 Potongan melintang badan kapal pada bagian buritan kapal



Gb.2.6. Potongan memanjang badan kapal

lebih dahulu dilakukan pencarian titik- titik yang diketahui yang mengurung titik yang dicari.

Jika harga y pada seluruh permukaan adalah nol , maka kurva pada sebarang harga u akan terbentuk garis lurus yang sejajar dengan sumbu x dan kurva pada sebarang harga w juga akan terbentuk garis lurus yang sejajar dengan sumbu z.

Hubungan antara x dan u tersebut ditunjukkan oleh gambar. , sedang hubungan antara z dan w ditunjukkan oleh gambar.....

Tampak pada kedua grafik tersebut bahwa hubungan antara u dengan x dan antara w dengan z mendekati bentuk linier. Dengan hubungan seperti ini maka metode iterasi yang

diambil adalah metode Newton – Raphson. Dengan metode ini harga yang dicari bisa diperoleh lebih cepat dan tidak ada kemungkinan terjadi jebakan terhadap penggunaan metode ini karena bentuk kurva yang mendekati linier tersebut.

BAB III.

BAHASA PEMROGRAMAN DAN HIDROSTATIK

III.1 Memahami bahasa delphi

Dasar bahasa pemrograman yang digunakan dalam Delphi adalah Pascal, sebuah bahasa yang didesain khusus oleh Niklaus Wirth untuk mengajarkan pemrograman terstruktur. Dibandingkan dengan bahasa generasi ketiga lainnya, seperti bahasa C, Pascal lebih mudah dipelajari dan digunakan. Hal ini karena Pascal memiliki struktur seperti bahasa Inggris sehingga mudah untuk dibaca . Disamping itu aplikasi yang dibuat dengan kompiler hasilnya hanya sebuah file EXE saja. Sampai saat ini belum ada lingkungan pengembangan yang menyediakan fasilitas ini.

Bahasa lain yang biasa dipakai oleh para pemula adalah BASIC (Beginner All Purpose Simbolic Instruction Code) . Bahasa ini kemudian dikembangkan dalam Visual Basic (VB) dari microsoft. Visual Basic merupakan produk yang mirip dengan Delphi namun memiliki fasilitas fasilitas berorientasi object yang berbeda. Meskipun VB juga menerapkan pemrograman event-driven dan telah membuktikan diri sebagai aplikasi pengembangan dalam windows yang profesional, namun VB tidak mempunyai perubahan berarti dalam fasilitas-fasilitas dasarnya.

Pascal dalam Delphi berbeda dengan Pascal pada versi-versi sebelumnya, bahkan bila dibandingkan dengan Borland Pascal versi 7. Objek Pascal dalam Pascal 7 merupakan pengembangan kompiler-kompiler Pascal versi sebelumnya. Sekarang dalam Delphi bentuk-bentuk dari objek Pascal lebih ditingkatkan lagi. Oleh Borland, Delphi

dikembangkan dengan tujuan menetapkan standar baru bahasa Pascal. Akan tetapi Delphi masih mampu mengenal bentuk-bentuk lama Objek Pascal dari versi kompiler yang sama.

III.2. Hidrostatik.

Kedua sembilan belas item yang terdapat dalam gambaran kurva-kurva hidrostatik akan dijelaskan melalui tabel berikut:

Tabel. 3.1.

No.	Nama Lengkung	Tanda	1 cm	Diukur dari station N
1	Luas Garis Air	WPA	M^2	0
2.	Displacement air laut	D	Ton	0
3.	Luas Permukaan Basah	WSA	M^2	0
4	Luas Midship	MSA	M^2	
5	Letak titik berat garis air terhadap penampang tengah kapal	Φ B	M	
6	Letak Titik Tekan terhadap Penampang Tengah	Φ F	M	
7	Letak Titik Tekan terhadap Keel	KB	M	0
8	Jari Jari Metacenter Melintang Kapal	TBM	M	0
9	Jari Jari Metacenter Memanjang Kapal	LBM	M	0
10.	Letak Metacenter Melintang	TKM	M	0
11.	Letak Metacenter Memanjang	LKM	M	0
12	Koefisien Garis Air	C_W		0
13	Koefisien Blok	C_B		0
14	Koefisien Gading Besar	C_M		0
15	Koefisien Prismatik Horizontal	Cp		0
16	Ton per 1 cm	TPC	Ton	0
17	Perubahan Displacement karena kapal mengalami trim buritan sebesar 1 cm	DDT	Ton	0
18	Momen untuk mengubah trim 1 cm	MTC	Tm	0

Keterangan :

1. Lengkungan Luas

   Lengkungan ini menunjukkan luas bidang garis air untuk tiap bidang garis air dengan satuan meter persegi.

2. Lengkung displacement di air laut.

   Displacement kapal dengan kulit di air laut [ton].

3. Lengkung Luas Permukaan Basah.

   Yaitu luas dari permukaan badan kapal yang berhubungan langsung dengan air laut[m2].

5. Luas Midship.

   Luasan pada midship kapal [m2]

6. Lengkung Titik Berat terhadap Penampang Tengah Kapal.

   Jarak titik berat garis air F (Flotation) terhadap penampang tengah kapal untuk berbagai keadaan sarat [m].

7. Letak Titik Tekan Terhadap midship kapal..

   Lengkung ini menunjukkan kedudukan titik tekan B terhadap penampang tengah kapal untuk pelbagai sarat [m].

8. Letak Titik Tekan terhadap Keel

   Lengkung ini menunjukkan letak titik tekan B terhadap Keel

9. Metacenter Melintang Kapal .
   Menunjukkan besarnya jari jari metacenter melintang kapal [m] .

10. Jari Jari Metacenter Memanjang Kapal.

    Menunjukkan besarnya jari jari metacenter memanjang kapal [m] .

11. Letak Metacenter Melintang.

    Lengkung ini menunjukkan letak metacenter melintang M terhadap keel pada pelbagai sarat air.

12. Letak Metacenter Memanjang

Menunjukkan letak metacenter memanjang terhadap keel untuk pelbagai sarat air.

13. Koefisien Garis Air

Rasio antara luas bidang garis air muat (WPA) dengan luas sebuah segiempat persegipanjang dengan panjang L dan lebar B.

14. Koefisien Blok

Ratio antara isi karene dengan isi suatu balok dengan panjang L, lebar B dan tinggi T.

15. Koefisien Gading Besar

Ratio antara luas penampang gading besar yang terendam air dengan luas penampang yang lebarnya B dan tingginya T.

16. Koefisien Prismatik Horizontal

Rasio antara volume badan kapal yang ada dibawah permukaan garis air (isi karene0 dengan volume sebuah prisma dengan luas Ax dan panjang L.

17. Ton per 1 cm
TPC adalah jumlah ton yang diperlukan untuk menyatakan perubahan sarat kapal sebesar 1 cm didalam air laut.

18. Perubahan Displacement karena kapal mengalami trim buritan sebesar 1 cm

Lengkung ini valid untuk kapal tidak pada kondisi trim, menjelaskan perubahan displacement kapal karena mengalami trim buritan sebesar satu centimeter

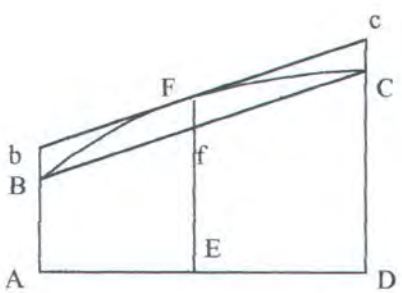
19. Momen untuk mengubah trim 1 cm

MTC lengkung ini menyatakan berapa besarnya momen untuk merubah kedudukan kapal dengan trim sebesar satu centimeter pada pelbagai sarat air.

III.3. Rumus Simpsons

Aturan aritmatika yang digunakan untuk mengukur suatu luas daerah yang salah satu sisinya adalah parabola dengan jarak ordinat yang sama dikenal sebagai rumus Simpsons, meskipun keasliannya ditemukan oleh Sir Isaac Newton. (....<no. ref...) .

Dasar metode Simpsons yang menurunkan sifat asosiatif pada sebuah parabola , seperti pada gambar berikut ini:



Gb.3.1 Dalil Simpsons I.

Dimana kurva BFC adalah tembereng parabola. Tangen bc pararel dengan panjang chord BC dan bertemu dengan parabola pada titik F. FE adalah ordinat tengah yang membagi rata garis AD.

Luas trapesium AbcD = EF x AD = 2EF x AE

Luas trapesium ABCD = (AB x DC) x AE

Kurva BFCf = 2/3 AD x f F = 2 m (pemisalan)

Luas BbFfC = 1/3 AD x f F = m

Dan Luas ABFCD = Area AbcD - m

= Area ABCD + 2 m

3 x Luas ABFCD =

(2 x Luas AbcD - 2m) + (Luas ABCD + 2m)

= (2 EF x AE) + [{ AB + DC) x AE]

$$= AE (4 EF + AB + DC)$$

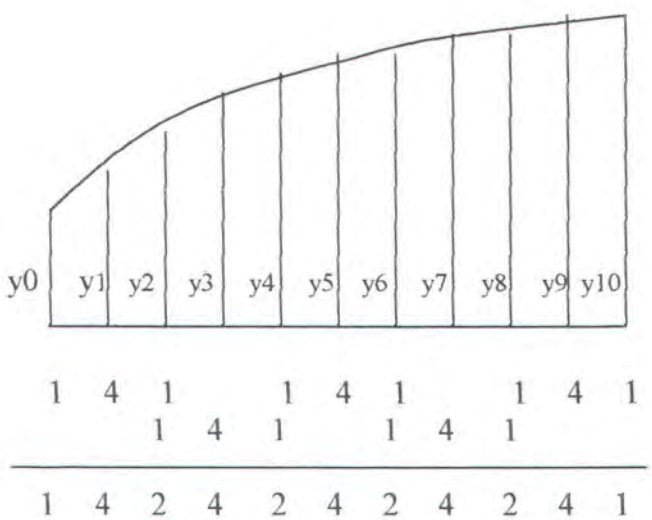
dengan notasi y dapat ditulis :

$$A = h/3 \ (Y_1 + 4Y_2 + Y_3)$$

Persamaan tersebut dikenal sebagai Rumus Simpsons atau Rumus Tiga Ordinat.

Rumus tersebut digunakan untuk menentukan luas dibawah kurva dengan ordinat yang jumlahnya ganjil dan jarak antar ordinatnya sama. Nilai awal pada koefisien Y yang disebut pengali-pengali kemudian digabung dengan yang lainnya seperti gambar berikut ini:

$$A = h/3 \ (Y_1 + 4Y_2 + 2Y_3 + 4Y_4 + 2Y_5 + 4Y_6 + 2Y_7 + 4Y_8 + 2Y_9 + 4Y_{10} + Y_{11})$$



Gb. 3.2. Rumus Simpsons I – Penggabungan pengali – pengali.

Diagram diatas adalah ½ dari garis air (WL) kapal maka untuk mencari keseluruhannya dikalikan dengan 2. Ini adalah sesuatu yang penting didalam keseluruhan kasus, baik ½

naupun satu bagian dari ordinat-ordinat yang dipakai. Untuk lebih jelasnya dengan melihat contoh dibawah ini :

Diketahui :

½ lebar kapal dengan tabel dan jarak antara ordinat 12,2 m adalah sebagai berikut :
Tabel. 3.2.

No. Ordinat	½ Lebar	Faktor Simpsons	Product Luas
0	2.0	1	2.0
1	7.3	4	29.2
2	9.8	2	19.6
3	10.4	4	41.6
4	10.6	2	21.2
5	10.7	4	42.8
6	10.6	2	21.2
7	9.9	4	39.6
8	7.8	2	15.6
9	4.2	4	16.8
10	0.2	1	0.2
		$\Sigma =$	249.8

Luas = 1/3 x 12,2 x 249,8 x 2 (untuk kedua sisi) = 2031,7 m^2

Momen dan titik berat.

Perhitungan didalam arsitektur perkapalan membutuhkan komputasi momen luas terhadap axis dan menentukan titik berat. Ini juga menentukan momen kedua dari luas / Momen Inersia terhadap axis yang dilalui titik beratnya.

Hubungan antara momen Luas , posisi titik berat dan luas adalah sebagai berikut :

Jarak titik berat dari axis yang diberikan = Momen Luas dari axis yang diberikan

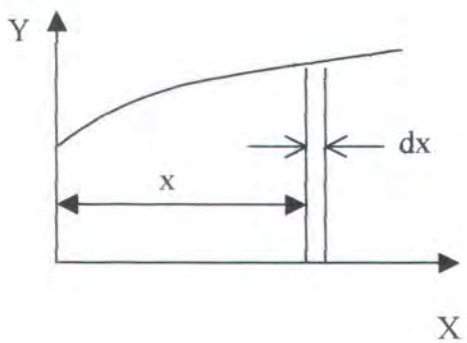
Area

Momen Luas membutuhkan beberapa sumbu , yaitu :

1. transversal / melintang
2. longitudinal / memanjang

1. titik berat luas dibawah kurva terhadap salah satu ordinatnya.

Pada gambar berikut ini , yaitu dengan mengambil elemen strip y dx dan momen dari strip terhadap ordinat OA adalah y x dx.



Gambar. . Ilustrasi dari elemen strip.

Momen Total Luas terhadap OA = $\int$ y x dx

dan

Total Luas = $\int$ y dx.

2. Titik berat terhadap base (garis dasar)

Dengan berasumsi bahwa elemen strip adalah persegipanjang dan titik beratnya $^{y}/_{2}$ dari base dan momen strip terhadap base adalah $\frac{1}{2} y^2 dx$.

Momen Total Luas terhadap base $= \int \frac{1}{2} y^2 dx$

dan

Total Luas $= \int y \, dx.$

Momen Luas dan posisi titik berat dengan menggunakan Rumus Simpsons I.

Contoh yang diberikan dibawah ini adalah garis air sebuah kapal, yang lazimnya untuk menyatakan posisi titik berat selalu diambil terhadap pertengahan kapal , misalkan diambil station midship . Lengan yang diukurkan dari posisi tersebut dan dengan sejumlah perhitungan aritmatic sederhana dan hasilnya dikalikan dengan interval / jarak antar ordinat . Jika perkalian momen mA (momen After) lebih besar dari mF (momen Fore) maka posisi titik berat terdapat pada buritan / belakang midship dan begitu juga sebaliknya jika mA lebih kecil dari mF maka titik beratnya terdapat didepan midship kapal.

Momen Inersia.

Momen inersia kedua dari luas atau juga dikenal dengan Momen Inersia Luas , adalah dengan menjumlahkan setiap elemen- elemen luasan dikali dengan jarak kuadrat elemen tersebut terhadap axis , seperti contoh berikut

Diketahui : ½ lebar kapal ,dengan jarak antar ordinat 12,2 m dijelaskan dengan tabel sebagai berikut :

Tabel.3.3.

Letak	No. Ord	½ Lebar	F S	Product Luas	Lengan terhadapΦ	Product Momen
A f t	**0**	2.0	**1**	2.0	5	10.0
	1	7.3	**4**	29.2	4	116.8
	2	9.8	**2**	19.6	3	58.8
	3	10.4	**4**	41.6	2	83.2
	4	10.6	**2**	21.2	1	21.2
midΦ	**5**	10.7	**4**	42.8	0	290.0 .. = mA
	6	10.6	**2**	21.2	1	21.2
	7	9.9	**4**	39.6	2	79.2
	8	7.8	**2**	15.6	3	46.8
	9	4.2	**4**	16.8	4	67.2
Forward	**10**	0.2	**1**	0.2	5	1.0
			Σ =	249.8		215.4 = mF
					Sisa =	74.6 aft

Luas Waterplane / Garis Air = $^1/_3$ x 12.2 x 249,8 x 2 (untuk kedua sisi) = 2031.7 m^2

Momen Area terhadap midship = $^1/_3$ x 12,2 x 12,2 x 74,6 x 2 (untuk kedua sisi)

= 7402.3 m^2m

Titik berat terhadap midship = $\frac{7402.3}{2031.7}$ = 3.64 m dibelakang midship

atau dengan cara lain

Titik berat terhadap midship = $\frac{74,6}{249,8}$ x 12,2 = 3,64 dibelakang midship

BAB IV

SUSUNAN PROGRAM

IV.1.Struktur dan Pembagian Menu Program.

Jika Program PMother dicompile akan ditemukan beberapa file di HARD DISK dengan ekstensi DPR , DFM, DCU, dan PAS dan EXE, dimana Ekstensi DPR yang adalah singkatan dari DELPHI PROJEC FILE, yang adalah Program utama, dan FILE ini disimpan dengan Format Teks (FILE ASCII). Yang berikutnya adalah EKSTENSI DFM kepanjangan dari DELPHI FORM untuk INISIALISASI PROPERTI pada komponen , misalnya ukuran FORM, teks pada button, memo, radio group, check Box dan sebagainya. File Ekstensi ini tidak bisa berdiri sendiri dan harus berpasangan dengan File Pas. Meskipun File DFM berbentuk File Biner, dapat dilihat isinya dengan IDE Delphi . Dan yang ketiga ekstensi PAS singkatan dari PASCAL. File ini berisi program sumber dari aplikasi anda dan karena program utama berada pada file DPR maka file pas berisi Deklarasi Unit.

Untuk ekstensi yang keempat yaitu Delphi Compiled Unit yang disingkat DCU , File ini adalah hasil kompilasi file pas, dapat dianalogikakan file DCU dengan file TPU (turbo Pascal Unit) Pada turbo pascal atau file OBJ pada C/C ++.. Dengan ekstensi yang terakhir file EXE, yang merupakan file Executable.

Semua dari kelima Ekstensi tersebut dapat disimpulkan bahwa file yang diperlukan untuk membuat sebuah aplikasi adalah DPR, DFM dan pas dapat dikatakan sebagai file

temporer yang akan digunakan untuk membangun file exe. Dengan demikian setelah file exe terbentuk maka sebaliknya file-file selain ketiga ekstensi TSB dihapus yang bertujuan untuk menghemat HARD DISK.

PENGOPERASIAN PROGRAM:

- Basknot.pas

  Unit ini berisi rutin-rutin untuk menghitung fungsi basis B-Spline dan knot vektor

- Bspl2D.pas

  Unit ini berisi rutin-rutin untuk menghitung B-Spline curve fitting melalui titik data

- Bsplfit.pas

  Unit ini berisi rutin-rutin untuk menghitung B-Spline surface fitting melalui titik data.

- CalcPoly.pas dan Calpol2.pas.

  Kedua unit ini berisi rutin-rutin untuk menghitung poligon melalui data parametris.

- CalcSurf.pas, smurf.pas dan Precasu.pas.

  Ketiga unit ini berisi rutin-rutin untuk menghitungsurface dan data poligon.

- IOData.pas

  Unit ini berisi rutin-rutin untuk keperluan input output suatu data.

- Mother.pas

  Unit ini berisi rutin-rutin untuk menampilkan jendela utama.

- Newton.pas dan newtonpr.pas.

  Unit ini berisi rutin-rutin untuk menghitungakar dengan metode newton-raphson.

- Unit UFind

Merupakan yang berisi subrutin untuk mencari nilai dari ke- 18 kurva pada berbagai macam sarat.

- Unit HitTabel

  Berisi sub rutin untuk menghitung ke-18 item kurva hidrostatik

- Unit CHidro

  Berisi sub rutin untuk menampilkan tabel pembagian station dan water line

- Unit UScale

  Berisi sub rutin untuk mengatur pen-skalaan dari 18 item kurva

Sedang program utama diberi nama HSH, singkatan dari Hull Surface Hidrostatik. Menu-menu dalam program HSH ini telah disusun secara terurut dari menu pemasukan data offset badan kapal hingga menu untuk menampilkan hasil potongan badan kapal dari persamaan permukaan. Secara lengkap akan diuraikan berikut ini.

VI. 1.1. Menu Pemasukan Data.

Dalam menu ini pemakai dapat memasukkan data offset badan kapal yang akan dicari persamaan permukaannya. Data Offset yang dimasukkan di sini berdasar garis air dan station yang ditentukan pemakaian. Selain data offset data lain yang harus dimasukkan adalah data lain yang harus dimasukkan adalah data bentuk haluan , yakni jarak linggi haluan dari station terakhir pada tiap garis air , juga data bentuk buritan yakni jarak linggi buritan dari station AP pada tiap garis air . Slain itu dalam program ini juga

dimungkinkan untuk memasukkan data kapal hingga Sheer jika diinginkan . Hasil pemasukan data ini kemudian akan disimpan dalam file dengan ekstension HFB.

Sub menu kedua dari menu ini adalah menu untuk menambah garis air pada data offset yang telah ada. Offset data badan kapal pada garis air tambahan ini akan dihitung sendiri oleh komputer dari data offset yang telah ada dengan teknik B-Spline curve fitting . Data offset yang telah ditambah ini juga akan disimpan dalam file berekstension HFB .

VI.1.2. MENU TAMPILAN 3 DIMENSI dari DATA Offset.

Untuk keperluan pengecekan bentuk data offset yang telah dimasukkan maka disediakan menu untuk menampilkan data offset tersebut dalam ruang tiga dimensi . Data-data offset yang ada dihubungkan dengan garis lurus sehingga terbentuk tampilan permukaan Wire Frame dari badan kapal.

VI.1.3. MENU GARIS AIR.

Dari data yang terletak di samping dalam file dapat dibentuk kurva-kurva garis airnya pada menu ini. Kurva-kurva ini adalah hasil bentukan dengan teknik B-Spline cure fitting dari data-data yang diambil pada tiap-tiap garis air , sebagaimana dibahas pada bab II .

Pada menu ini pula dapat disusun data parametris dari data asli pada arah parameter u dengan teknik sebagaimana dibahas pada bab III , sehingga akan

terbentuk station parametris . Pada pembentukan ini sekaligus akan dilakukan penggabungan dari data offset dengan data bentuk haluan dan bentuk buritan . Bentuk garis air dari hasil data parametris juga akan ditampilkan untuk mengecek jika terjadi penyimpangan bentuk garis air aslinya . Hasil dari data parametris dapat dismpan dalam file dengan ekstensional HFB pula . dan akan diberi kode tertentu sehingga program akan mengenalinya sebagai data offset parametres pada arah u pada saat file ini dibaca . File hasil ini juga dapat dibuka pada menu kedua untuk dilihat bentuk data parametris ini pada ruang 3 dimensi.

VI.1.4. MENU BODY PLAN .

Pada menu ini akan dibentuk kurva-kurva body line dari data yang telah tersimpan dalam file . Kurva-kurva disini juga merupakan hasil bentukan dengan teknik B-Spline cure fitting .

. Penyusunan data parametris pada arah parameter W dapat dilakukan pada menu ini dengan teknik yang sama dengan pada menu ketiga . Bentuk body line dari data parametris juga akan ditampilkan untuk pengecekan . Data yang dapat disusun kembali sebagai data parametris pada arah W selain data asli juga data yang telah tersusun secara parametris pada arah U hasil dari menu ketiga sehingga akan terbentuk data parametris pada kedua arah parametris.

VI.1.5. Menu penghitungan Poligon Permukaan l

Pada menu ini dilakukan perhitungan poligon permukaan dari data yang telah tersimpan dalam file dengan menggunakan teknik B-Spline Surface fitting

yang telah dibahas pada bab II . Semua tipe data file dengan ekstension HFB baik data asli dari menu pertama atau data parametris hasil dari menu ketiga dan keempat dapat dibentuk jaring-jaring poligon permukaannya dengan menggunakan teknik B-Splinesurface fitting . Poligon hasil perhitungan ini disimpan dalam file dengan ekstension PLG.

VI.1.6. Menu Penghitungan Titik –titik Permukaan

Dari data poligon yang telah diperoleh dapat dibentuk permukaan B-Spline nya pada menu ini dengan cara sebagaimana telah dijelaskan pada bab II. Dari hasil data-data

BAB V

PENGUJIAN PROGRAM

Pengujian program dilaksanakan untuk memperoleh ketepatan perhitungan, dimana dalam hal ini ada tiga buah jenis data. Ketiga jenis data tersebut adalah data setengah bola. Data pertama adalah hasil perhitungan hidrostatik dengan menghitung secara analitis, yang kedua adalah perhitungan dengan memakai tabel atau tabulasi yang lazim dipakai oleh mahasiswa perkapalan, dan yang terakhir adalah perhitungan dengan memakai program hidrostatik ini. Untuk menjelaskan perbedaan antara hasil data yang satu dengan hasil data yang lainnya penulis disini memakai Standar Deviasi yang akan memberikan gambaran tentang penyimpangan hasil data yang satu dengan yang lainnya, atau lebih tepatnya perbandingan antara hasil perhitungan hidrostatik tabulasi dan hasil perhitungan hidrostatik dengan program dengan hasil perhitungan hidrostatik secara analitis.

Hasilnya adalah seperti tabel dibawah ini.

No.	Standar Deviasi	Tabulasi	Program
1	WPA	0.44613517	**0.00392717**
2	Displasemen	2.69442608	**0.02110695**
3	WSA	6.70229990	**0.79970586**
4	MID F	0.00000000	0.00000000
5	MID B	0.00000000	0.00000000
6	KB	0.01359413	**0.00000198**
7	TBM	0.11426478	**0.00016215**
8	TKM	0.12034683	**0.00016164**
9	LBM	0.11414528	**0.00061548**
10	LKM	0.12081821	**0.00061726**
11	Cw	0.01738216	**0.00004696**
12	Cb	0.00863541	**0.00003974**
13	Cm	0.01422902	**0.00002672**

14	Cp	0.01189671	**0.00003931**
15	TPC	0.00489810	**0.00016380**
16	DDT	0.00000000	**0.00000000**
17	MTC	3.10795139	**0.03528235**
18	MSA	0.30193932	0.00115615

Maka dapat disimpulkan bahwa ke-15 kurva dari ke-15 kurva (tidak termasuk Φ B, Φ F dan DDT yang sama bernilai nol) adalah lebih teliti dengan menggunakan program ini, dan bukan itu saja dari segi efisiensi paling tidak program ini akan lebih singkat dalam pengerjaannya.

V.1. Keterbatasan program.

Program yang telah disusun dalam Skripsi ini memiliki beberapa keterbatasan sehingga tidak dapat digunakan untuk semua jenis badan kapal . Keterbatasan ini timbul dikarenakan keterbatasan metode B-Spline dalam mendefinisikan permukaan. Program ini tidak dapat membentuk permukaan yang diskontinyu. Permukaan diskontinyu dimaksud adalah permukaan yang mengandung suatu titik atau bidang yang kontinyu pada turunan tertentu , antara lain seperti knucle.

Knucle atau sudut tajam pada lambung kapal , banyak dijumpai antara lain pada kapal cepat yang menggunakan chine, pada kapal-kapal dengan skeg dibagian buritannya dan juga pada kapal yang mempunyai buritan berbentuk transom.

V.2. Kekurangan program

Program Hidrostatik ini disusun masih mempunyai beberapa kekurangan. Yang dimaksud dengan kekurangan adalah karena hal-hal tersebut dapat diatasi dengan menambah maupun memodifikasi program yang telah ada ini .

Kekurangan kekurangan ini antara lain:

- Tidak tersedianya fasilitas untuk mencetak gambar dari program. Dalam hal ini tidak tersedianya fasilitas ekspor data gambar ke format grafik standart yang akan sangat berguna untuk software yang lain.

  Pencetakkan tampilan bukanlah hal yang vital dalam program ini, karena program ini ditujukan untuk keperluan analisa dan bukan untuk keperluan desain atau produksi sebagaimana pada program untuk keperluan design dan produksi.
- Kemampuan analisa dari permukaan yang telah dibentuk dengan program ini , hanya terbatas pada pembentukkan potongan-potongan badan kapal yang berupa station, garis air dan buttock line.

BAB VI

KESIMPULAN DAN SARAN

VI.1. Kesimpulan

Penulis menyimpulkan bahwa dari pengujian dan perbandingan program ini jika dibandingkan dengan tabulasi terhadap perhitungan analitis, didapatkan ketelitian yang bervariasi yaitu dimulai dari yang paling terkecil standart deviasinya yaitu KB sebesar 0.00000198 kemudian yang menengah yaitu TBM, TKM, LBM, LKM , Cb, Cp, Cm, Cw,berkisar antara 0.0002 s/d 0.0006, dan dilanjutkan yang paling besar standart deviasinya yaitu WPA, Displacemen dan WSA yaitu berkisar antara 0.03 s/d 0.7, dan didalam ilmu statitistik nilai standart deviasi $\pm$ 1 merupakan nilai yang kecil, yang berarti nilai-nilai itu kurang berserakan bahkan lebih consentrated.

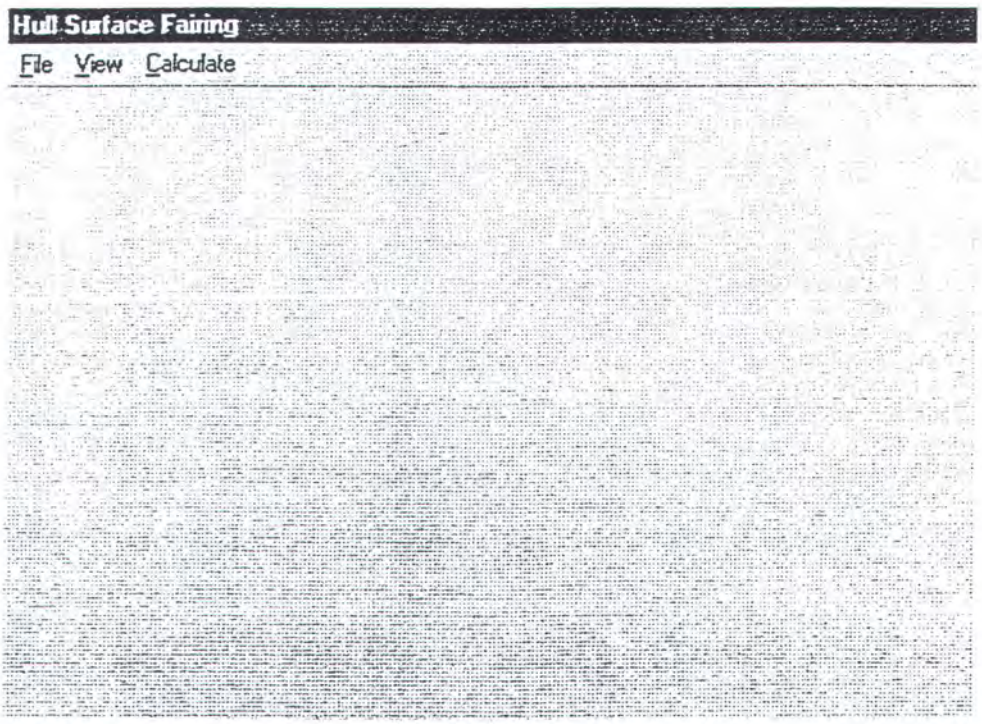
VI.2. Saran Pengembangan.

Penulis pernah mencoba untuk pembagian sebanyak 101 station dan 101 water line, yang kemudian ditingkatkan menjadi 191 baik untuk station maupun water line-nya. Diharapkan untuk yang kedepan, maka pembagian dapat dibesarkan lagi misalkan menjadi 301 ataupun 501 baik untuk station maupun water line-nya, untuk menganalisa keakurasian data hasil perhitungan hidrostatik, yang tentunya juga ditunjang dengan kemampuan internal komputer itu sendiri, misalnya saja processor .

Program ini barulah program pra-permulan untuk perhitungan hidrostatik, yang didalamnya belum menyertakan kemampuan analisa data dan manipulasi permukaan yang dibentuk. Sehingga diharapkan adanya kelanjutan dan keterkaitan untuk memodifikasi bahkan mengembangkan program yang berhubungan dengan hidrostatik ini.

LAMPIRAN A : PETUNJUK PENGGUNAAN PROGRAM

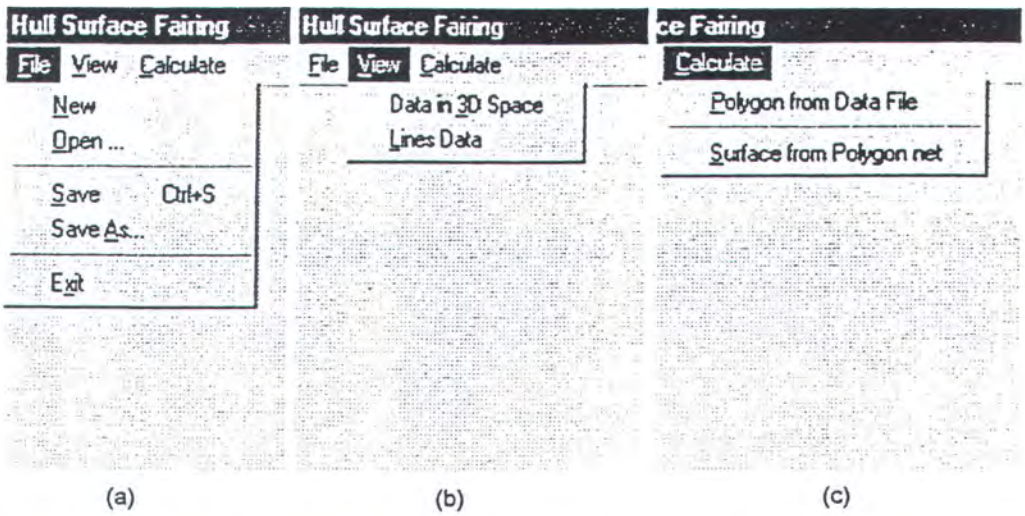
Setelah program dieksekusi (dijalankan) maka di layar akan terlihat jendela utama seperti pada gbr.A.1 . Jendela utama mempunyai menu utama File, View dan Calculate.



Gbr. A.1 Jendela utama program

Menu utama File mempunyai sub-menu New, Open, Save as, Save dan Exit seperti terlihat pada gbr.A.2.(a). Sub-menu New dipilih jika ingin membuat dan memasukkan data baru. Pemasukan data baru dimulai dengan mengisikan data jumlah station dan water line kemudian dilanjutkan dengan mengisi data half breadth.

Sub-menu Open berguna untuk membuka file data yang telah tersimpan dalam media penyimpan (hard disk atau disket).

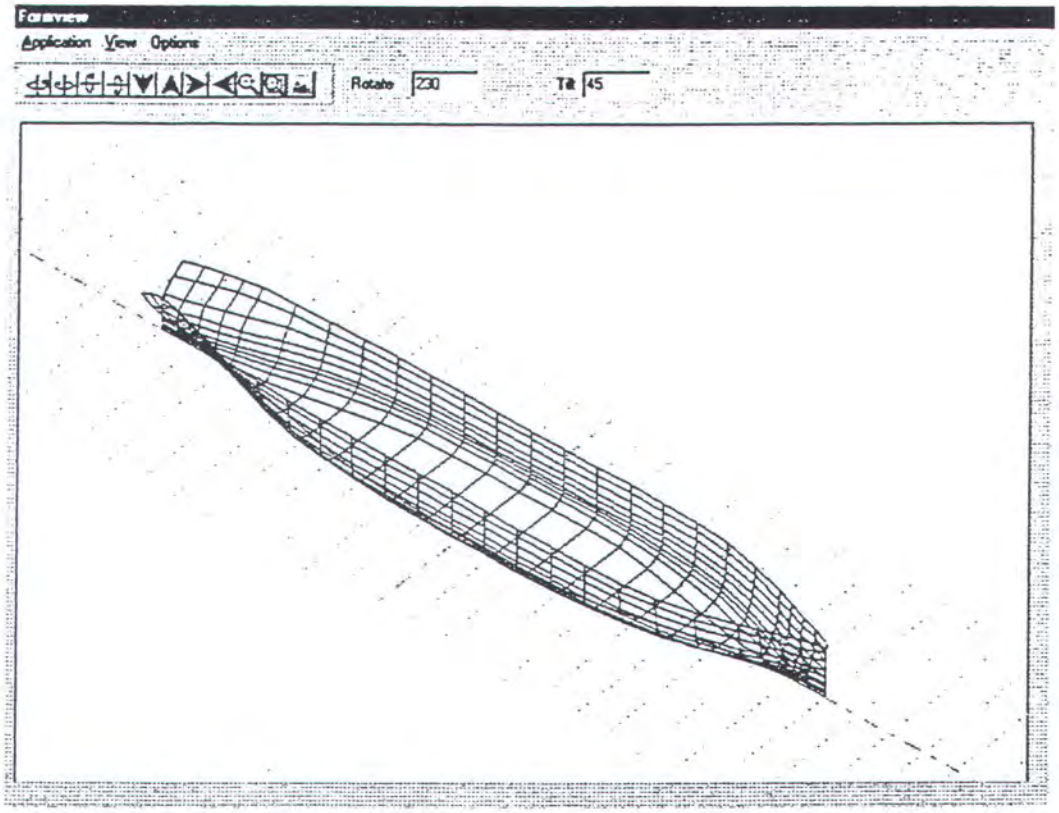


Gbr. A.2 Pembagian menu utama. (a). Menu File dan sub-menunya. (b) Menu View dan sub-menunya. (c). Menu Calculate dan sub-menunya.

Sub-menu Save as dan Save berguna untuk menyimpan data ke dalam media penyimpan. Sub-menu Save akan menyimpan data ke dalam nama file data yang sedang aktif, sedangkan sub-menu Save as akan menanyakan nama file tempat data akan disimpan. Sub-menu Exit digunakan untuk keluar dari program.

Menu utama View mempunyai dua sub-menu yaitu Data in 3D Space dan Lines seperti terlihat pada gbr.A.2.(b). Jika salah satu sub-menu ini dipilih maka akan ditampilkan jendela View seperti yang terlihat pada gbr.A.3. Jendela View memiliki 3 menu utama dan 11 bit button . Menu utama terdiri dari Application, View dan Options.

Menu utama jendela View application terdiri dari sub-menu Print dan sub-menu Exit. Sub-menu Print digunakan untuk mencetak gambar dengan printer, sedang sub-menu Exit dipakai untuk keluar dari jendela View.



Gbr.A.3 Jendela View

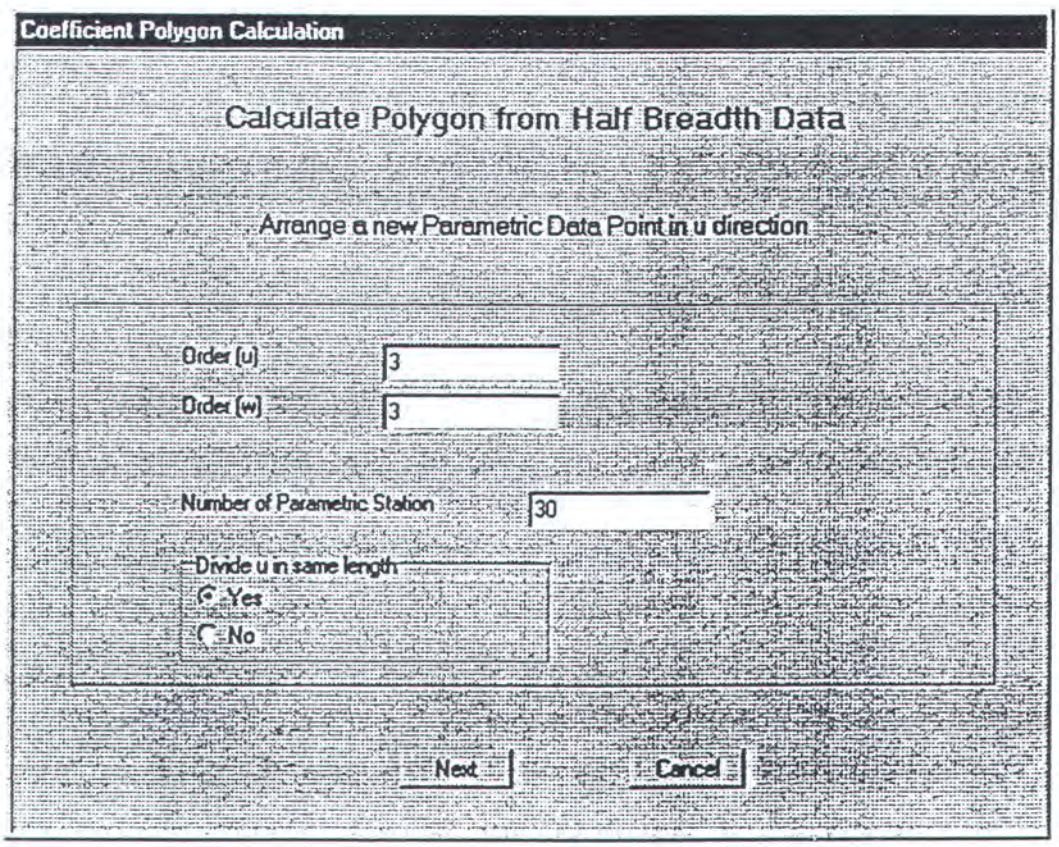
Menu utama jendela View View terdiri dari sub-menu Body Plan, Water Line, Buttock Line, Gaussian Curvature dan Polygon Vertices. Semua sub-menu ini dipakai untuk menampilkan dan menghilangkan body plan, Water Line, Buttock Line, Gaussian Curvature dan Polygon Vertices.

Bit button 1 dan 2 berguna untuk memutar gambar kapal dengan sumbu tegak kapal sebagai sumbu putar, sedang bit button 3 dan 4 berguna untuk memutar gambar kapal dengan sumbu memanjang kapal sebagai sumbu putar. Bit button 5, 6, 7 dan 8 digunakan untuk menggerakkan gambar sesuai dengan arah panah yang ditunjukkan bit button. Bit button 9 dan 10 digunakan untuk

memperkecil dan memperbesar gambar. Bit button 11 digunakan untuk menampilkan gambar setengah kapal atau penuh.

Menu utama Options terdiri dari sub-menu Select Polygon dan change poligon. Sub-menu select digunakan untuk memilih suatu daerah tertentu dari gambar. Setelah daerah tertentu dipilih, maka akan ditampilkan beberapa poligon yang berpengaruh pada daerah yang dipilih. Dengan sub-menu change pemakai program dapat memilih polygon yang akan dirubah.

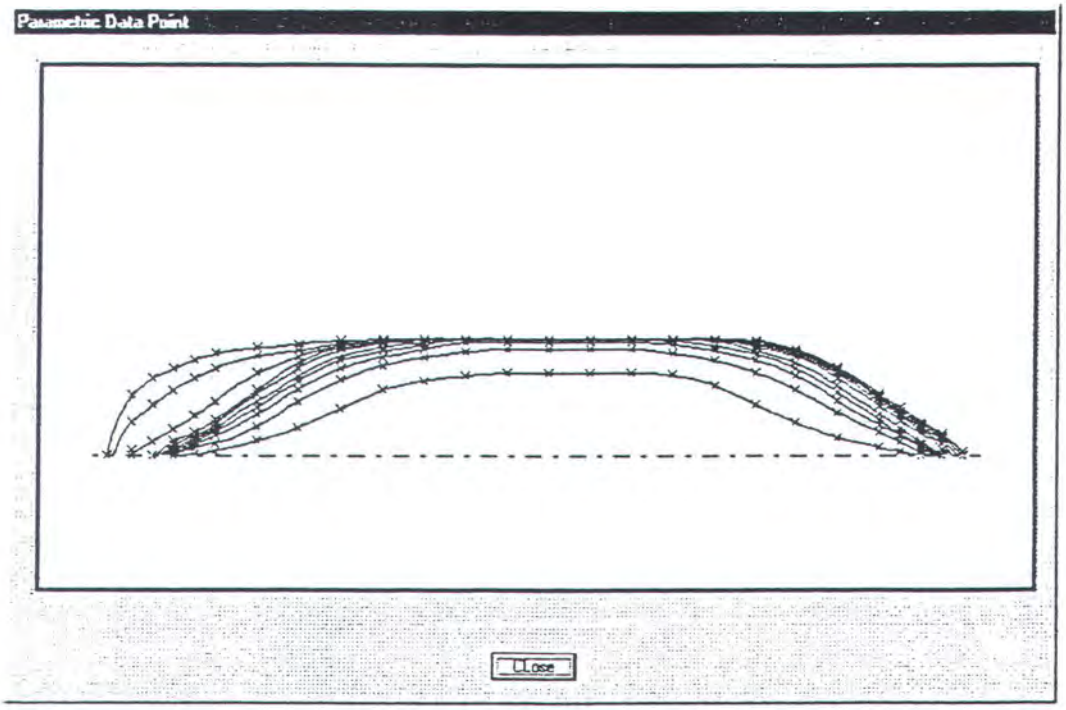
Menu utama Calculate dari jendela utama terdiri dari dua sub-menu yaitu Polygon from Data File dan Surface from Polygon seperti terlihat pada gbr.A.2.(c). Sub-menu from Data File dipilih jika pemakai program akan menghitung poligon dari data setengah lebar kapal. jika sub-menu ini dipilih maka



Gbr.A.4 Jendela perhitungan data parametris

akan ditampilkan jendela perhitungan data parametris seperti ditunjukkan gbr.A.4. Jendela ini berfungsi untuk menerima input yang akan dipakai dalam perhitungan data station parametris. Pada jendela ini terdapat tiga kotak input yaitu order (u) ,order (w) dan jumlah station parametris serta dua button yaitu button Next dan button Cancel. Button Next digunakan jika akan melanjutkan perhitungan sementara button Cancel untuk membatalkan perhitungan. Setelah kotak input diisi dan dipilih button Next maka akan ditampilkan jendela gambar data station parametris seperti terlihat pada gbr.A.5. Jendela ini hanya berfungsi untuk memperlihatkan gambar data station parametris karena itu hanya mempunyai satu button yaitu button Close untuk menutup jendela. Setelah jendela gambar data station parametris ditutup maka akan ditampilkan kembali jendela perhitungan data parametris seperti pada gbr.A.4 tetapi kotak input jumlah station parametris dirubah menjadi jumlah water line parametris.. Jendela ini berfungsi untuk menerima input untuk perhitungan data water line parametris. Setelah button Next dipilih maka akan ditampilkan kembali jendela pada gbr.A.5 tetapi gambarnya merupakan data waterline parametris. Jika jendela ini ditutup maka akan ditampilkan jendela perhitungan poligon seperti terlihat pada gbr.A.6. Jendela ini mempunyai dua button yaitu button Next untuk melanjutkan perhitungan dan button Cancel untuk membatalkan perhitungan. Jika dipilih button Next maka perhitungan poligon dimulai dan kemajuan proses perhitungan akan ditunjukkan dengan persen dalam kotak progress seperti terlihat pada gbr.A.7.

Jika perhitungan poligon telah selesai maka akan ditampilkan kotak Save dan pemakai program dapat memilih file untuk menyimpan data poligon.

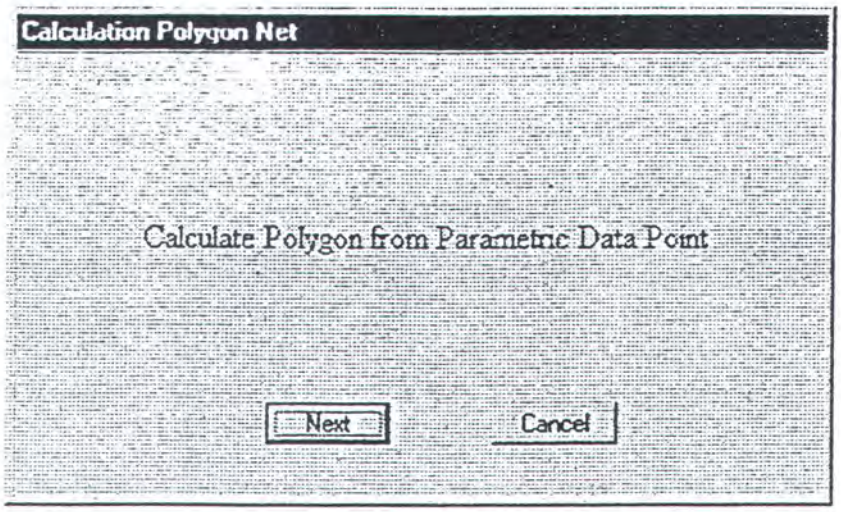


Gbr A.5 Data station parametris

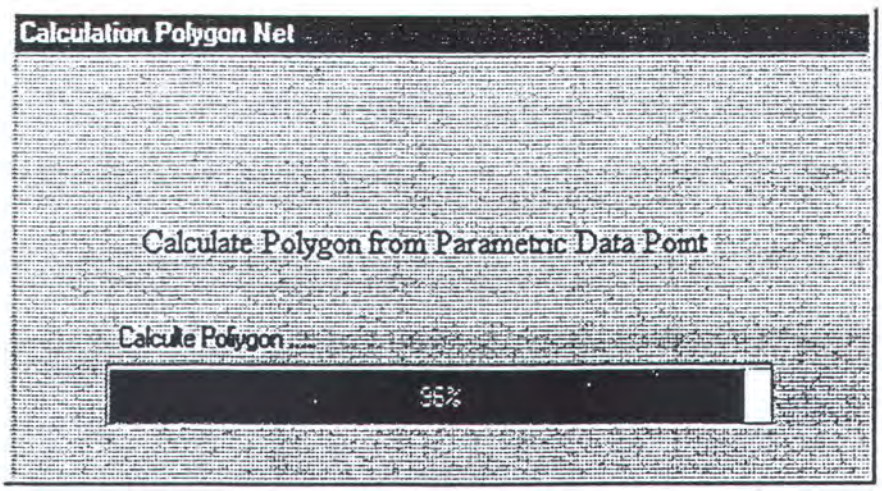
Sub menu Surface from Polygon pada jendela utama dipilih jika pemakai program akan menghitung permukaan dari file data poligon. Jika sub-menu ini dipilih maka akan ditampilkan kotak Open dan pemakai program dapat memilih file data poligon yang akan dipakai untuk perhitungan permukaan. Setelah file data poligon dipilih maka akan ditampilkan jendela perhitungan permukaan seperti terlihat pada gbrA.8.

Jendela perhitungan permukaan mempunyai 5 button. Button pertama adalah button Transverse Section (Body Plan). Button ini dipakai untuk melihat gambar salah satu potongan melintang (body plan). Jika button ini dipilih maka akan ditampilkan jendela Transverse Section seperti pada gbr.A.9.

Jendela Transverse Section mempunyai dua kotak input. Kotak input pertama adalah jarak ujung belakang kapal ke potongan melintang yang akan digambar, sedang kotak input kedua merupakan pembagian potongan melintang.



Gbr.A.6 Jendela Polygon Calculation

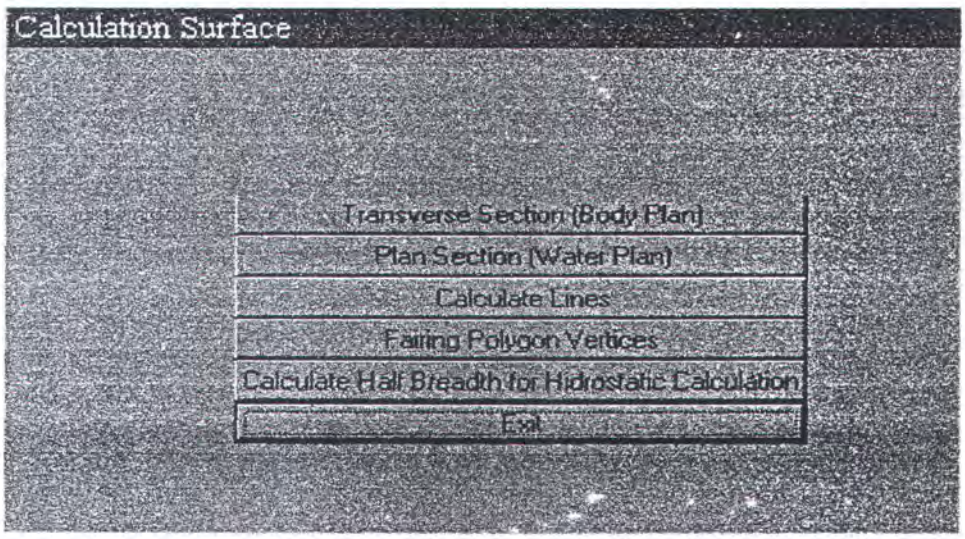


Gbr.A.7 Jendela Polygon Calculation dengan kotak kemajuan proses

Jendela Transverse Section juga memiliki dua button yaitu button OK untuk melihat gambar dan button Close untuk kembali ke jendela perhitungan permukaan.

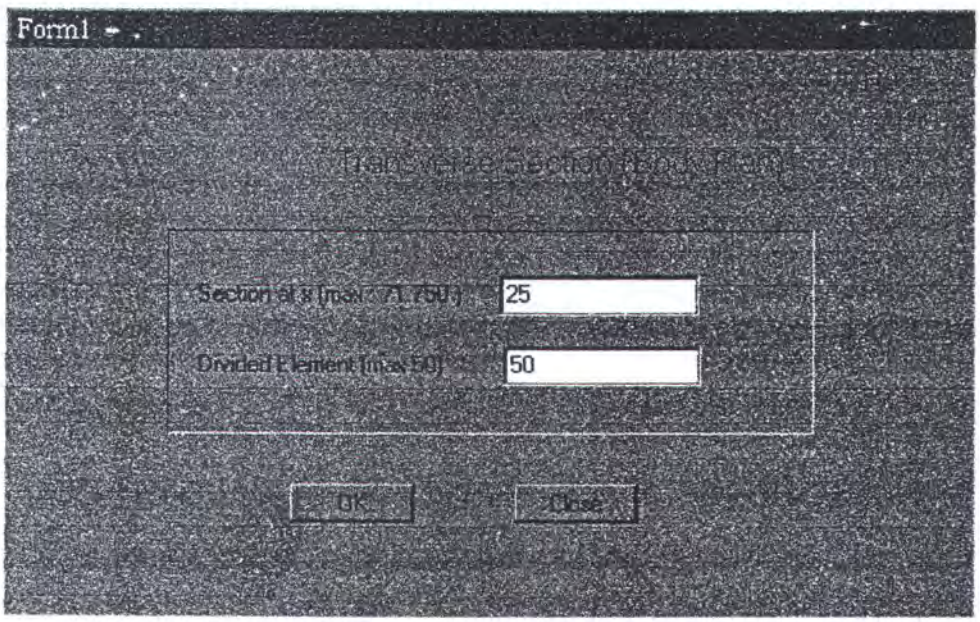
Button kedua dari jendela perhitungan permukaan adalah button Plan Section (Water Plan). Jika pemakai program memilih button ini maka akan ditampilkan jendela Plan Section. Bentuk jendela ini sama dengan bentuk jendela Transverse Section.

Button ketiga dari jendela perhitungan permukaan adalah button Calculate Lines yang berfungsi untuk menghitung lines dari kapal. Jika button ini dipilih maka akan ditampilkan jendela yang meminta input jumlah Station , jumlah Water Line dan jumlah Buttock Line. Setelah input ini dimasukkan maka akan ditampilkan jendela kemajuan perhitungan lines. Setelah perhitungan Lines selesai maka akan ditampilkan kotak Save dan pemakai



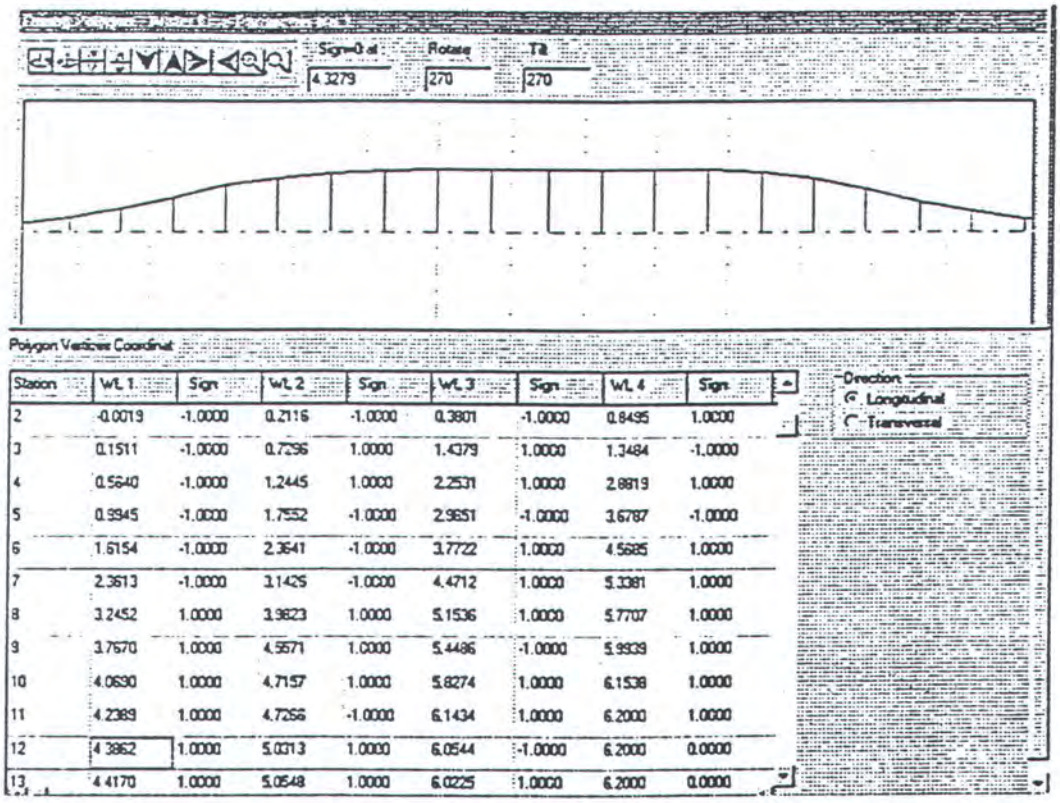
Gbr. A.8. Jendela perhitungan permukaan

program dapat memilih File untuk menyimpan data Lines



Gbr.A.9. Jendela Transverse Section

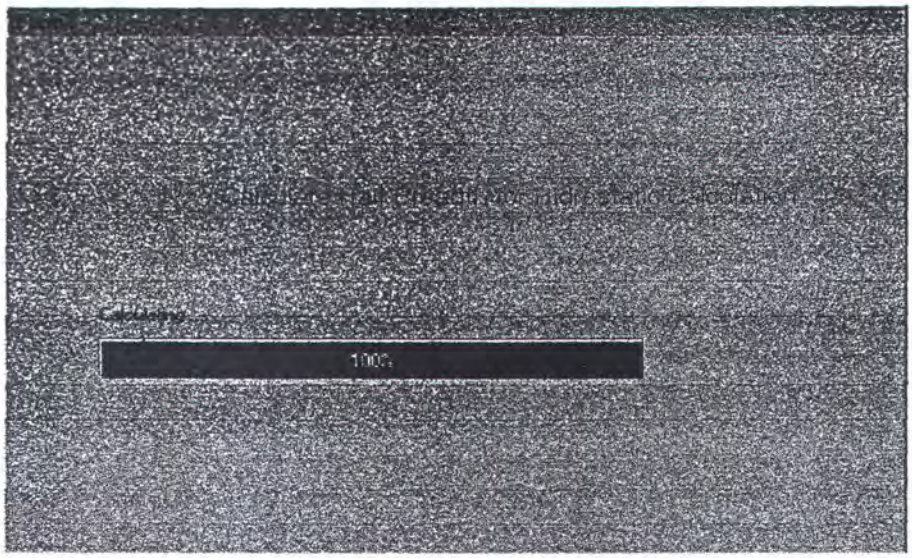
Button keempat dari jendela perhitungan permukaan adalah button Fairinf Polygon Vertices yang berfungsi sebagai proses fairing polygon. Jika button ini dipilih maka akan ditampilkan jendela Fairing Polygon . Bentuk jendela ini adalah seperti gbr.A.10. Proses Fairing Manual dapat dilakukan dengan mengedit langsung nilai koordinat polygon pada tabel gbr.A.10



Station	Wt. 1	Sign	Wt. 2	Sign	Wt. 3	Sign	Wt. 4	Sign
2	-0.0019	-1.0000	0.2116	-1.0000	0.3801	-1.0000	0.8495	1.0000
3	0.1511	-1.0000	0.7236	1.0000	1.4379	1.0000	1.3484	-1.0000
4	0.5640	-1.0000	1.2445	1.0000	2.2531	1.0000	2.8819	1.0000
5	0.9345	-1.0000	1.7552	-1.0000	2.9651	-1.0000	3.6787	-1.0000
6	1.6154	-1.0000	2.3641	-1.0000	3.7722	1.0000	4.5685	1.0000
7	2.3613	-1.0000	3.1425	-1.0000	4.4712	1.0000	5.3381	1.0000
8	3.2452	1.0000	3.9823	1.0000	5.1536	1.0000	5.7707	1.0000
9	3.7570	1.0000	4.5571	1.0000	5.4486	-1.0000	5.9939	1.0000
10	4.0630	1.0000	4.7157	1.0000	5.8274	1.0000	6.1538	1.0000
11	4.2389	1.0000	4.7256	-1.0000	6.1434	1.0000	6.2000	1.0000
12	4.3862	1.0000	5.0313	1.0000	6.0544	-1.0000	6.2000	0.0000
13	4.4170	1.0000	5.0548	1.0000	6.0225	1.0000	6.2000	0.0000

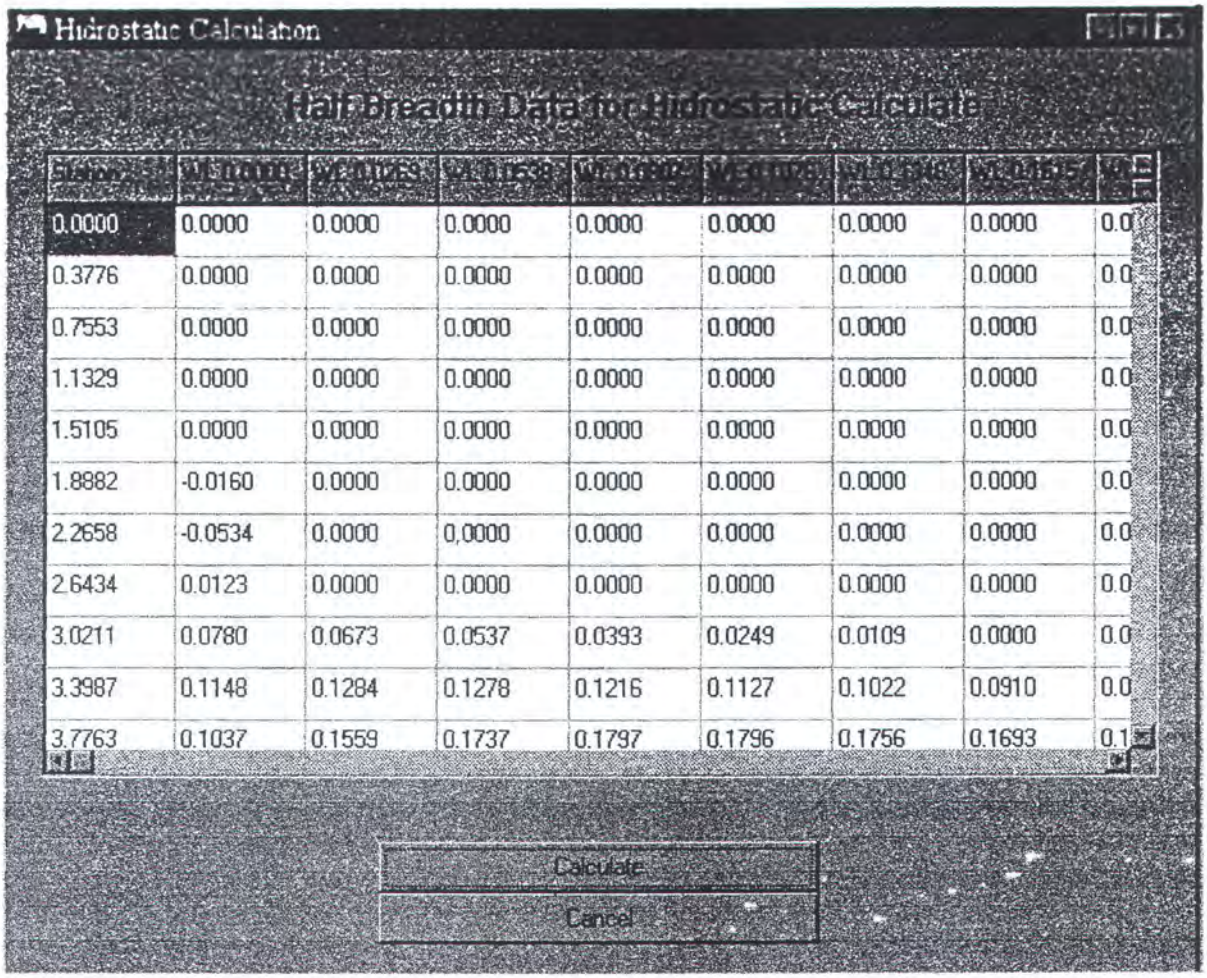
Gbr.A.10. Jendela Fairing Polygon

Button kelima dari jendela perhitungan permukaan adalah button Calculated Half Breadth for Hidrostatic Calculation. Dimana jika button ini dipilih akan ditampilkan seperti gbr.A.11



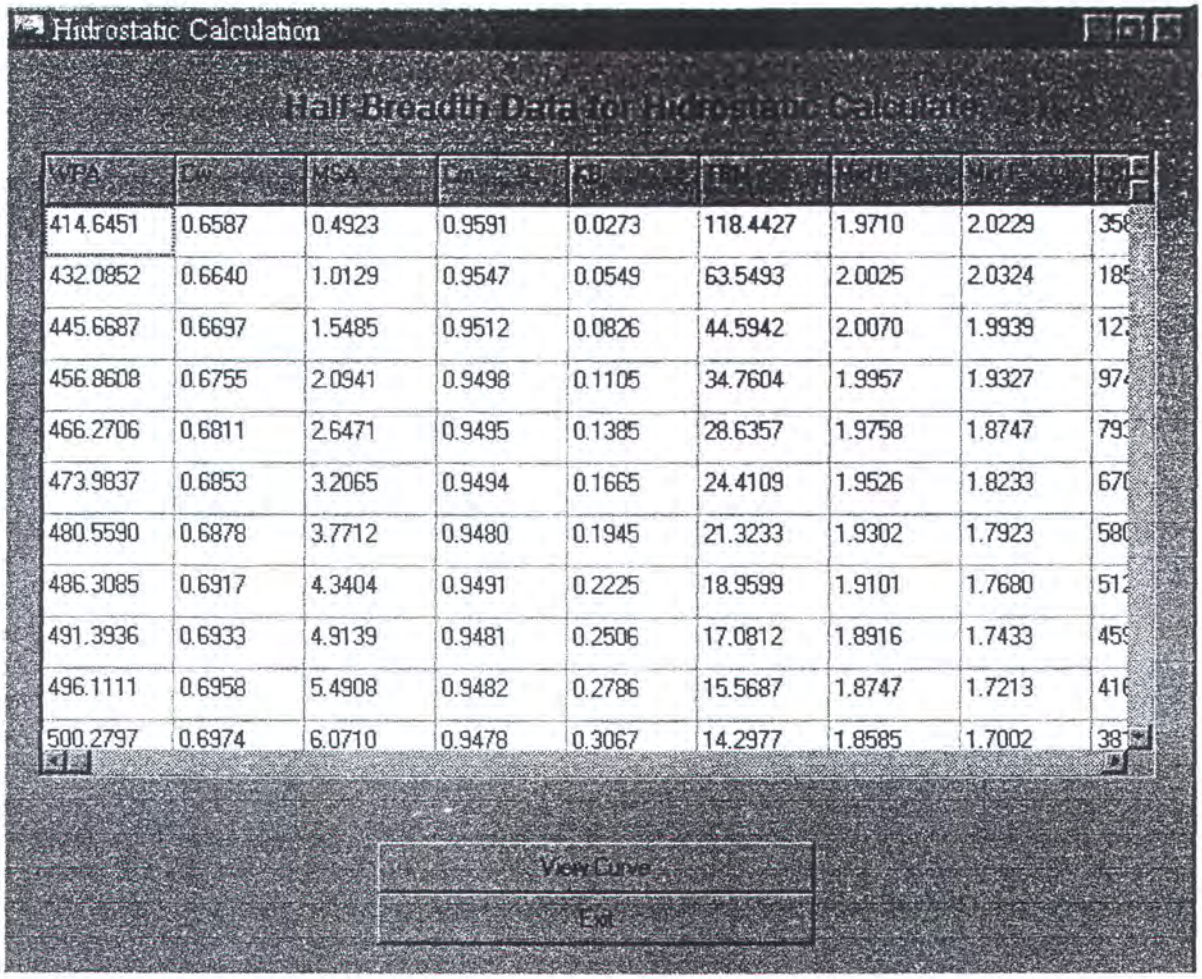
Gbr.A.11. Jendela Kemajuan Perhitungan Hidrostatik

Data ini kelak akan dipergunakan untuk perhitungan hidrostatik pada Sub-menu utama Calculate yaitu Hidrostatic.Jika Hidrostatic ini dieksekusi maka akan ditampilkan jendela yang berupa tabulasi half breadth yang telah diproses, seperti yang ditunjukkan pada gbr.A.12.



gbr.A.12 Tabulasi half breadth yang telah diproses

Setelah Button Calculate dipilih , maka tampilan berikutnya seperti pada gbr.A.13.,yaitu hasil perhitungan hidrostatik yang berupa tabel untuk setiap item kurva

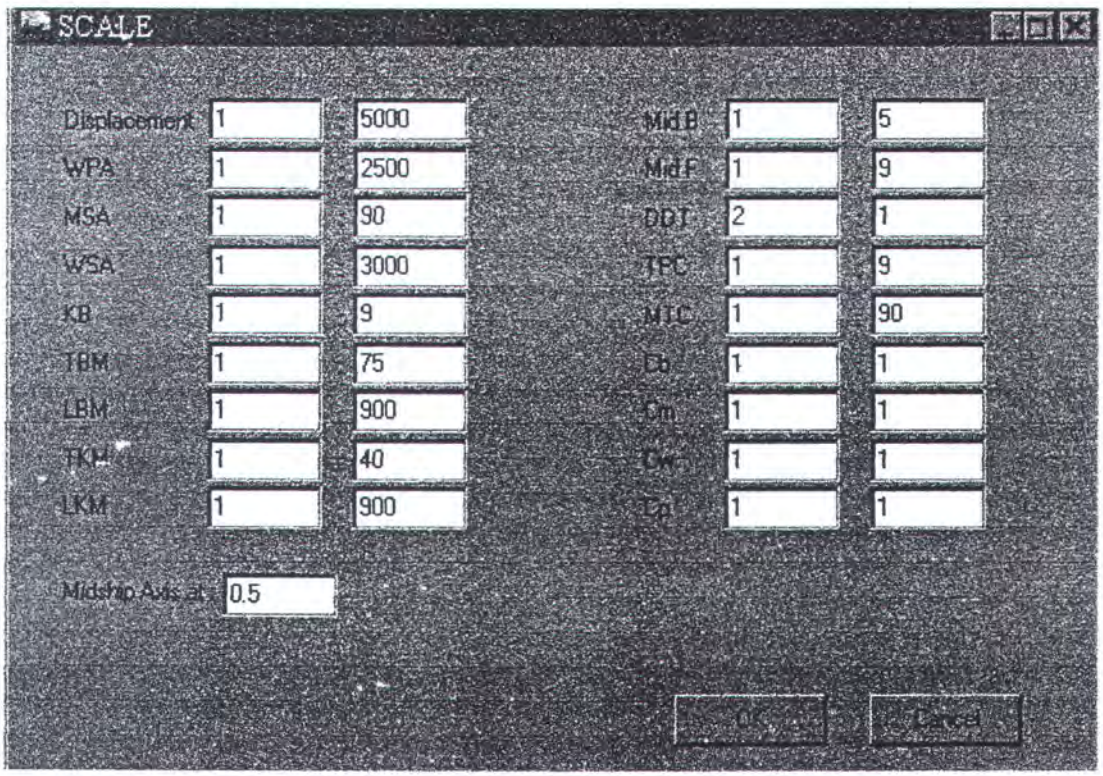


WPA	Cw	MSA						
414.6451	0.6587	0.4923	0.9591	0.0273	118.4427	1.9710	2.0229	358
432.0852	0.6640	1.0129	0.9547	0.0549	63.5493	2.0025	2.0324	185
445.6687	0.6697	1.5485	0.9512	0.0826	44.5942	2.0070	1.9939	12
456.8608	0.6755	2.0941	0.9498	0.1105	34.7604	1.9957	1.9327	97
466.2706	0.6811	2.6471	0.9495	0.1385	28.6357	1.9758	1.8747	79
473.9837	0.6853	3.2065	0.9494	0.1665	24.4109	1.9526	1.8233	67
480.5590	0.6878	3.7712	0.9480	0.1945	21.3233	1.9302	1.7923	58
486.3085	0.6917	4.3404	0.9491	0.2225	18.9599	1.9101	1.7680	51
491.3936	0.6933	4.9139	0.9481	0.2506	17.0812	1.8916	1.7433	45
496.1111	0.6958	5.4908	0.9482	0.2786	15.5687	1.8747	1.7213	41
500.2797	0.6974	6.0710	0.9478	0.3067	14.2977	1.8585	1.7002	38

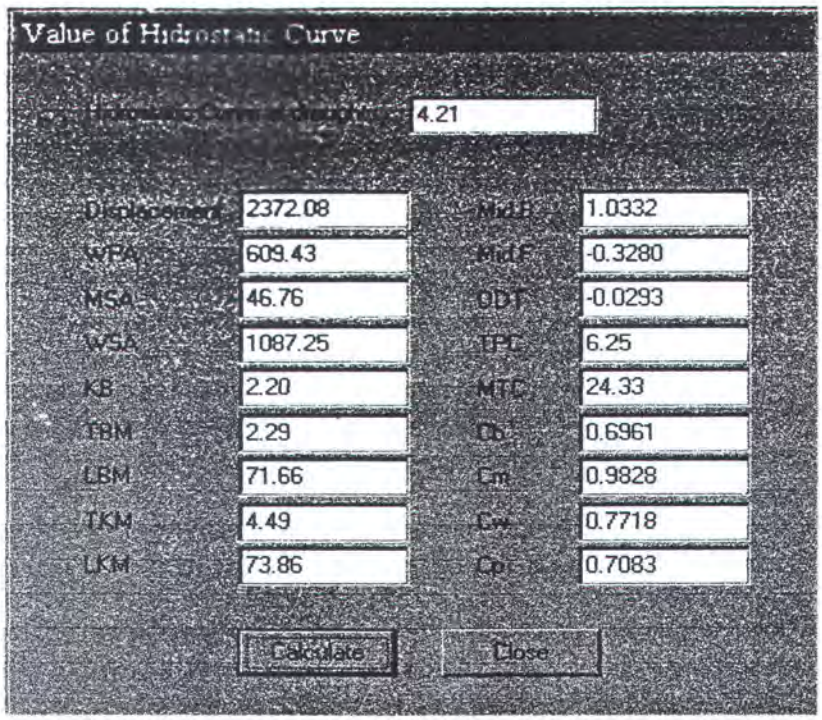
gbr.A.13 Hasil perhitungan hidrostatik yang berupa tabel

Dengan memilih button View Curce , maka akan ditampilkan pada layar seperti gbr.A.14.

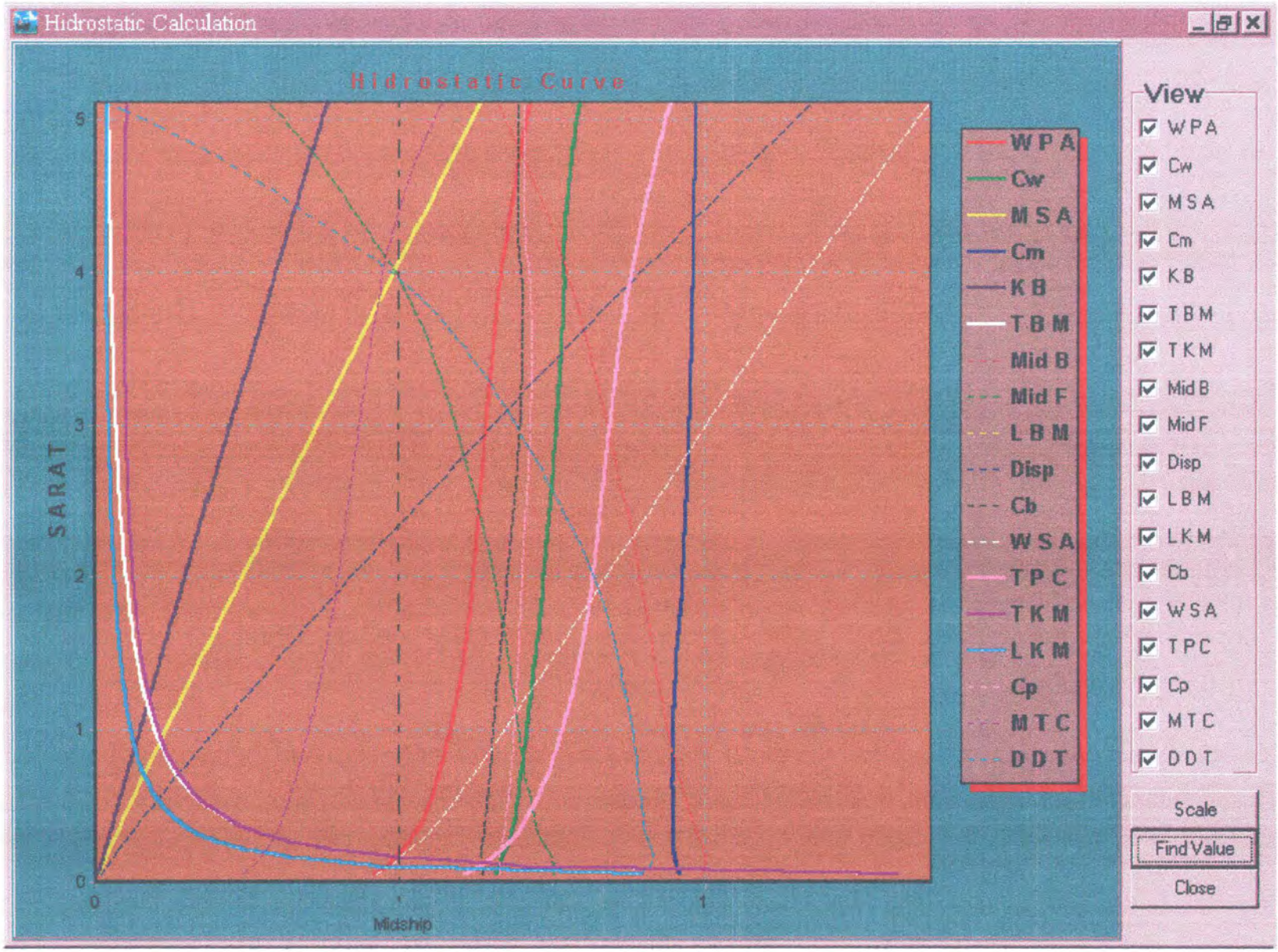
Pemilihan button scale pada jendela hidrostatic adalah untuk mengatur skala dari setiap item kurva, dan button Find Value berfungsi untuk mengetahui ke-18 item kurva hidrostatik pada pelbagai sarat kapal. Hal tersebut ditunjukkan berturut-turut pada gbr.A.15 dan 16.

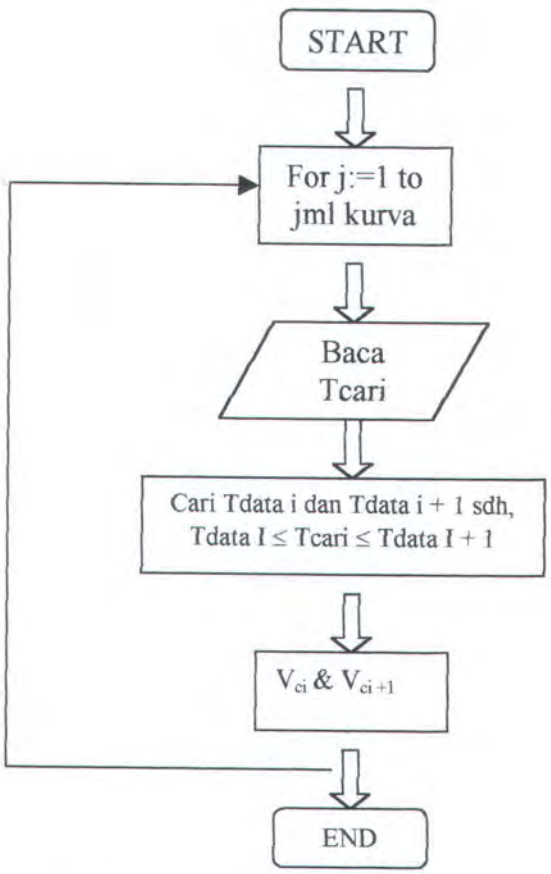


gbr.A.15. Jendela untuk mengatur skala ke-18 item kurva hidrostatik.

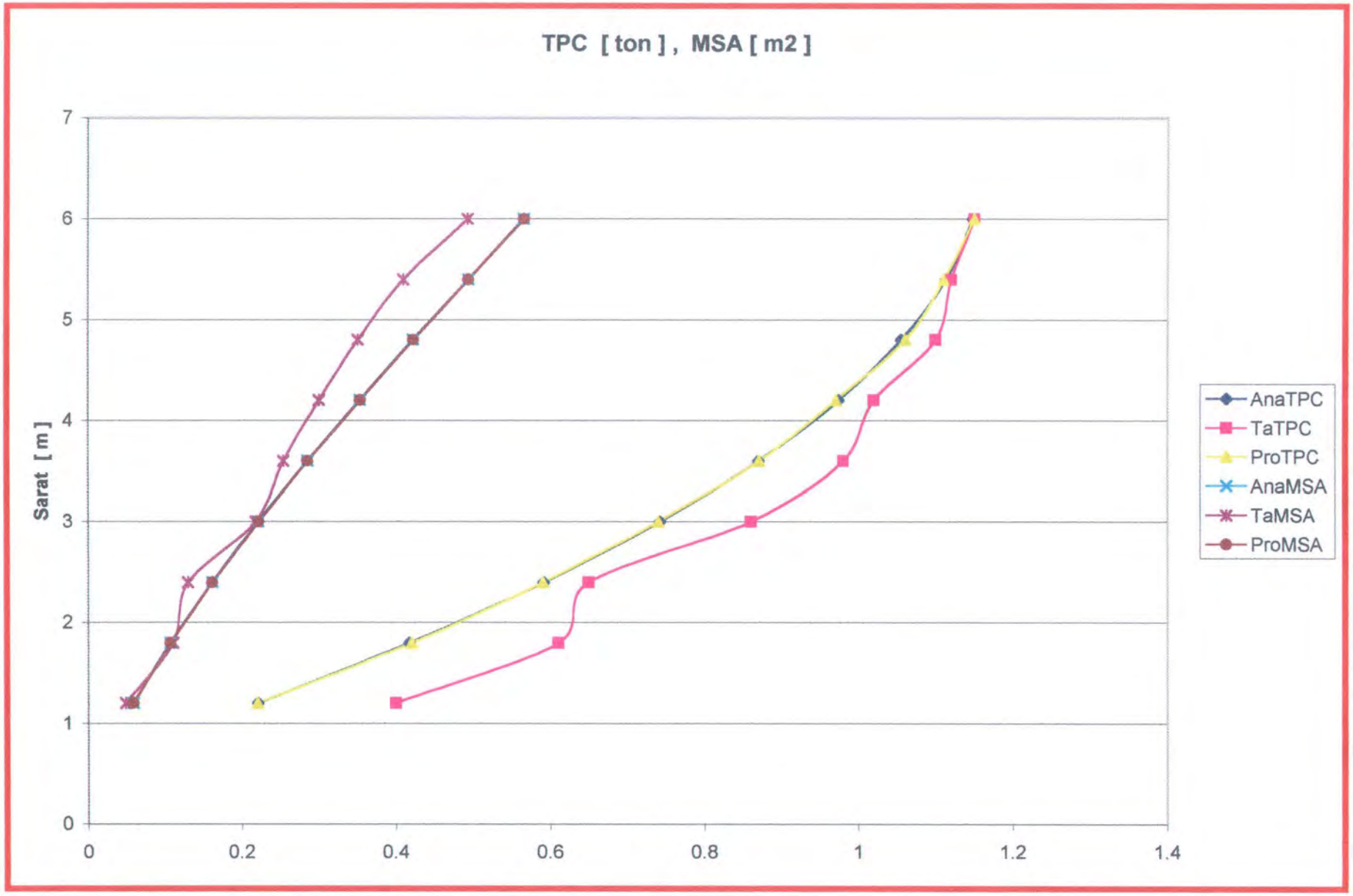


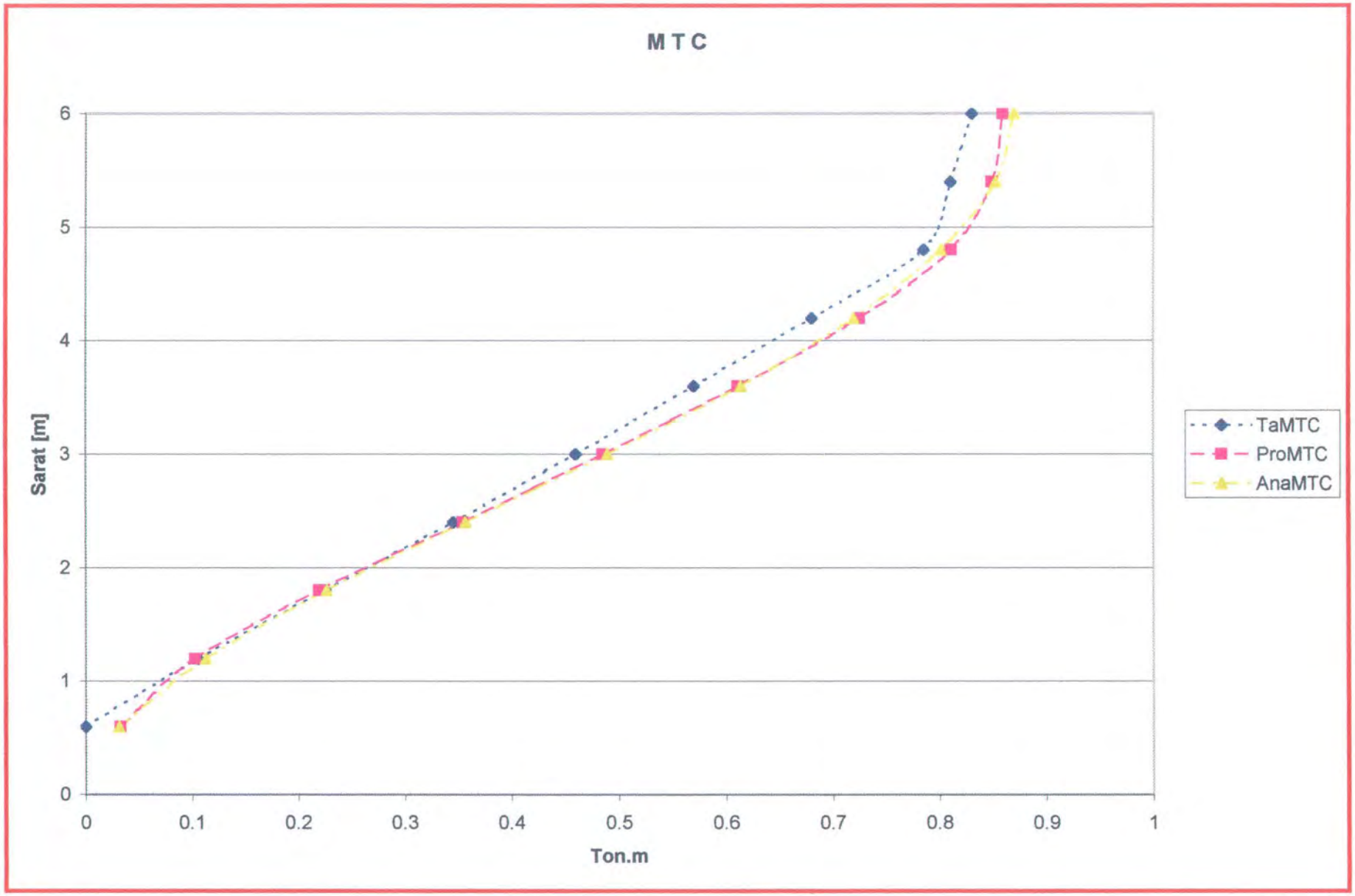
gbr.A.16. Kurva Hidrostatik pada pelbagai sarat kapal

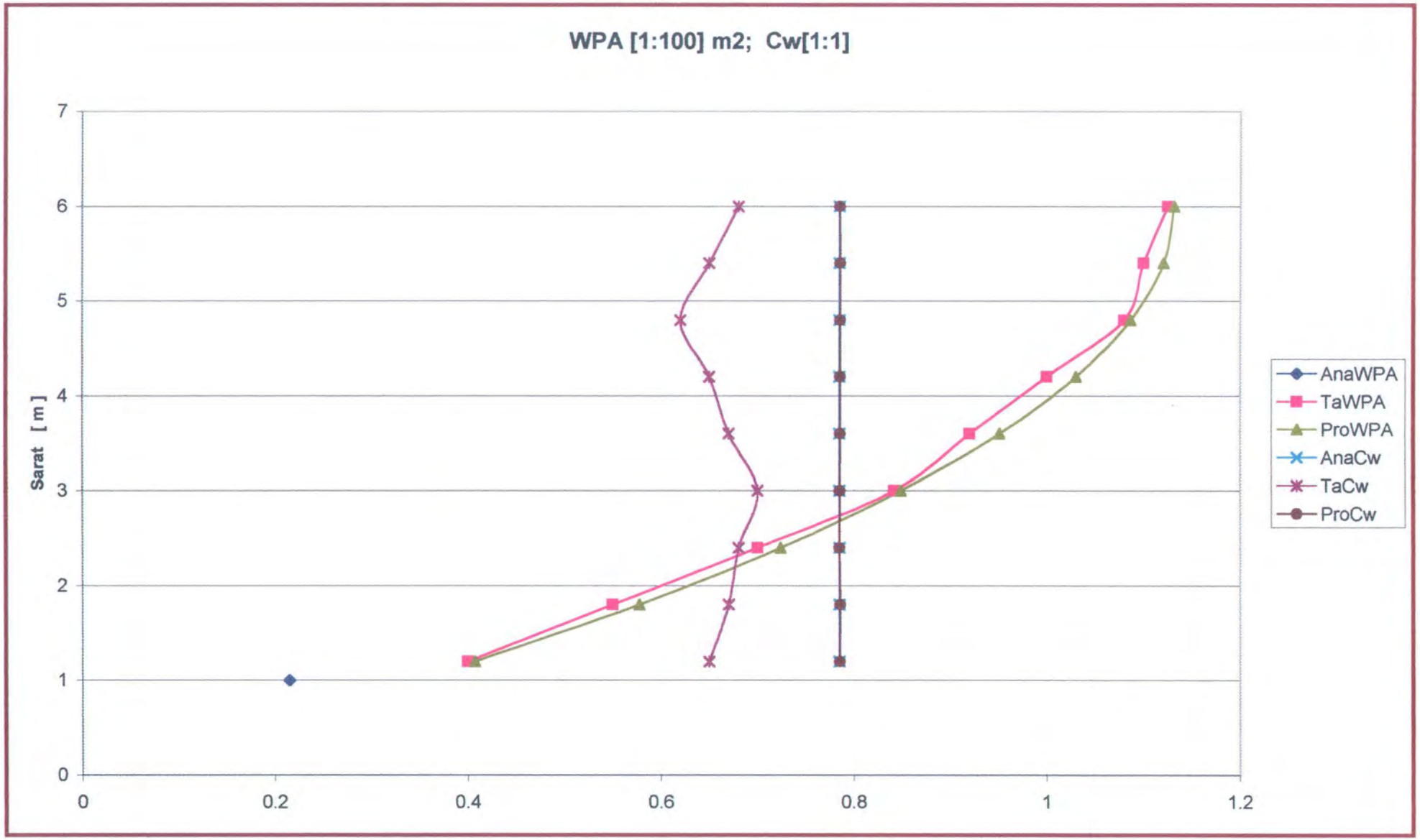

Hidrostatic Calculation
Hidrostatic Curve
SARAT
Midship
0
1
2
3
4
5
0
1
WPA
Cw
MSA
Cm
KB
TBM
Mid B
Mid F
LBM
Disp
Cb
WSA
TPC
TKM
LKM
Cp
MTC
DDT
View
WPA
Cw
MSA
Cm
KB
TBM
TKM
Mid B
Mid F
Disp
LBM
LKM
Cb
WSA
TPC
Cp
MTC
DDT
Scale
Find Value
Close

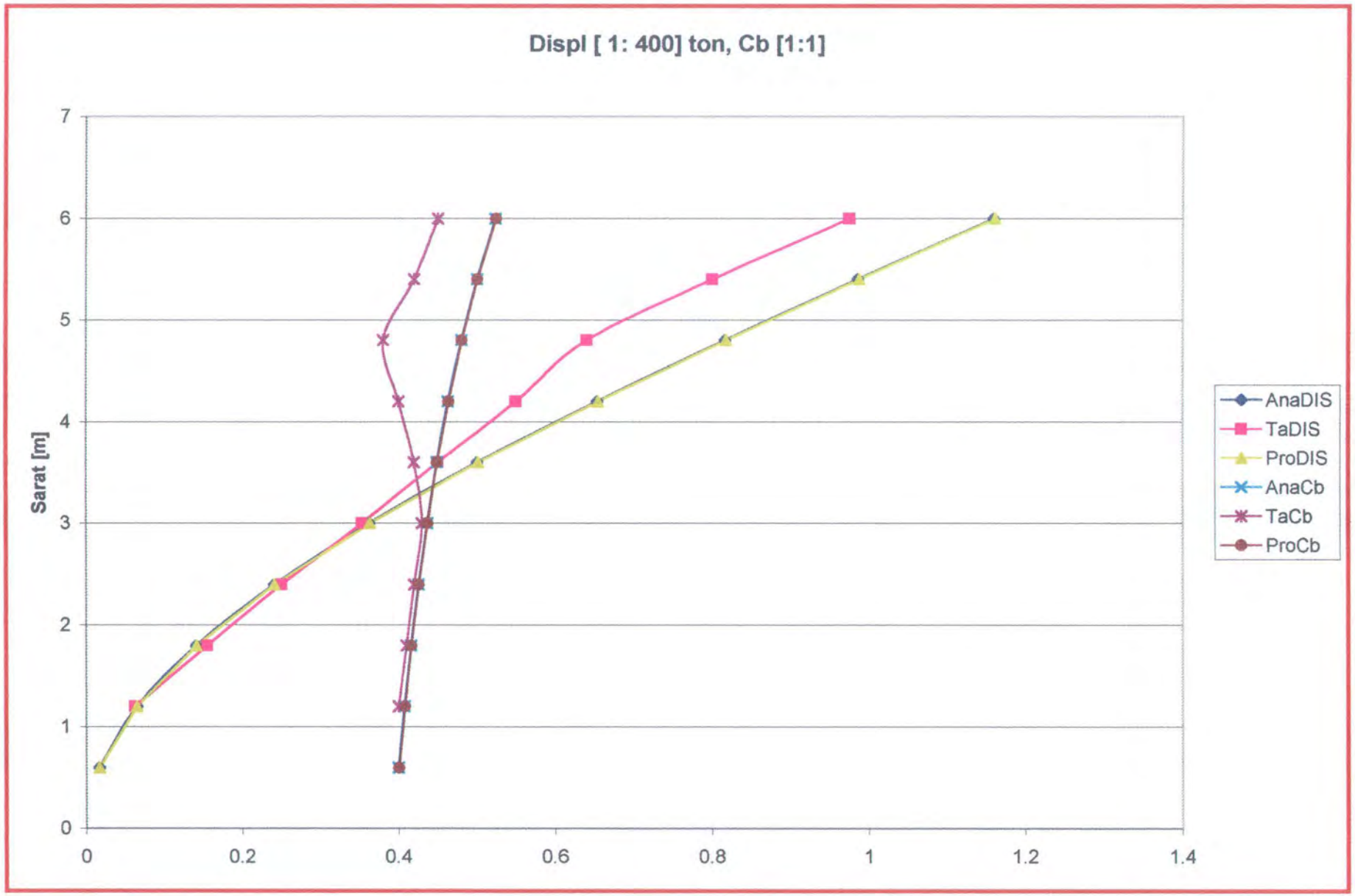


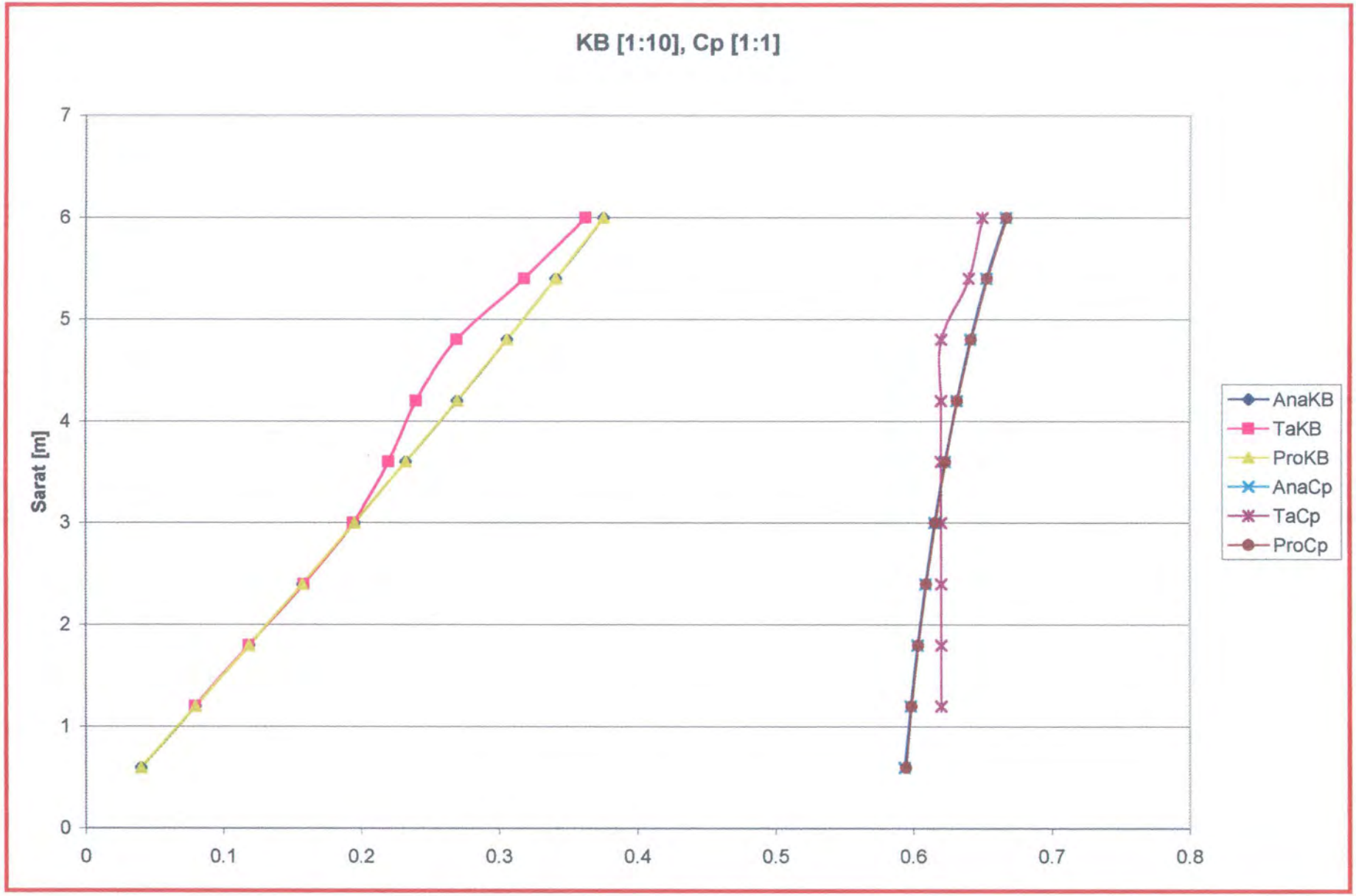
Gambar Flow Chart mencari ke-18 item kurva hidrostatik pada pelbagai sarat

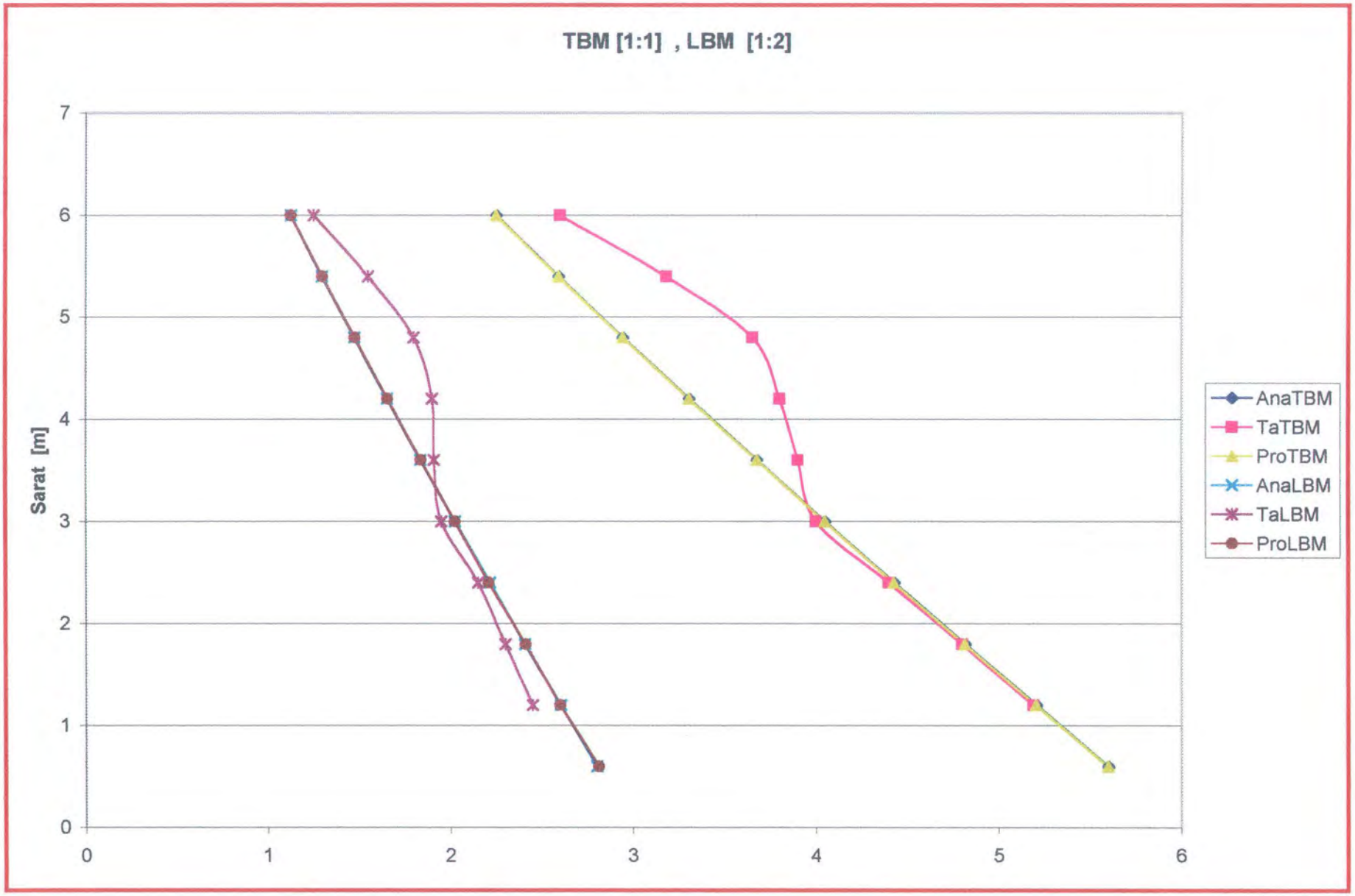

TPC [ton], MSA [m2]
Sarat [m]
0
1
2
3
4
5
6
7
0
0.2
0.4
0.6
0.8
1
1.2
1.4
AnaTPC
TaTPC
ProTPC
AnaMSA
TaMSA
ProMSA

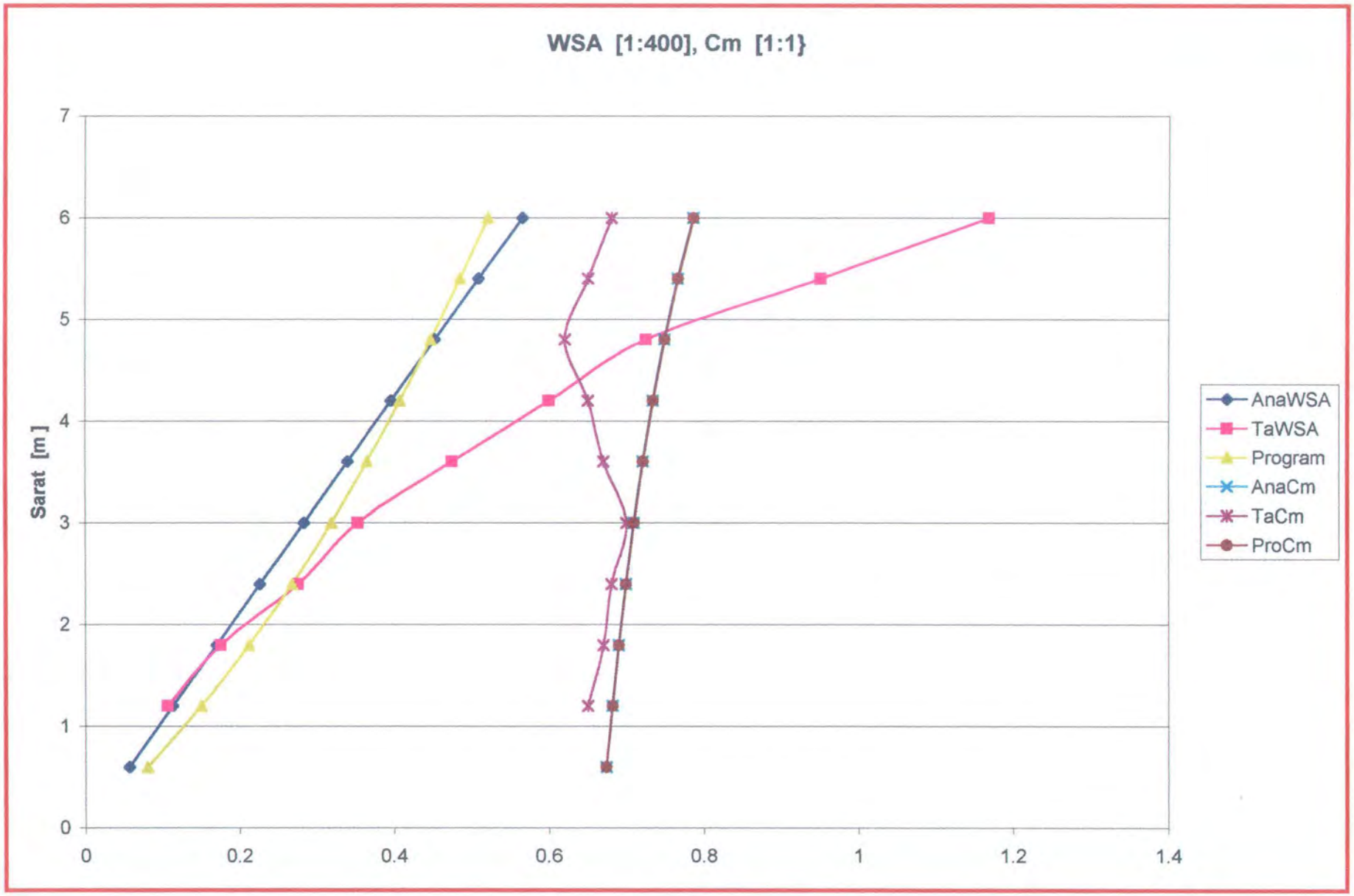

MTC
Sarat [m]
Ton.m
0
1
2
3
4
5
6
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
TaMTC
ProMTC
AnaMTC

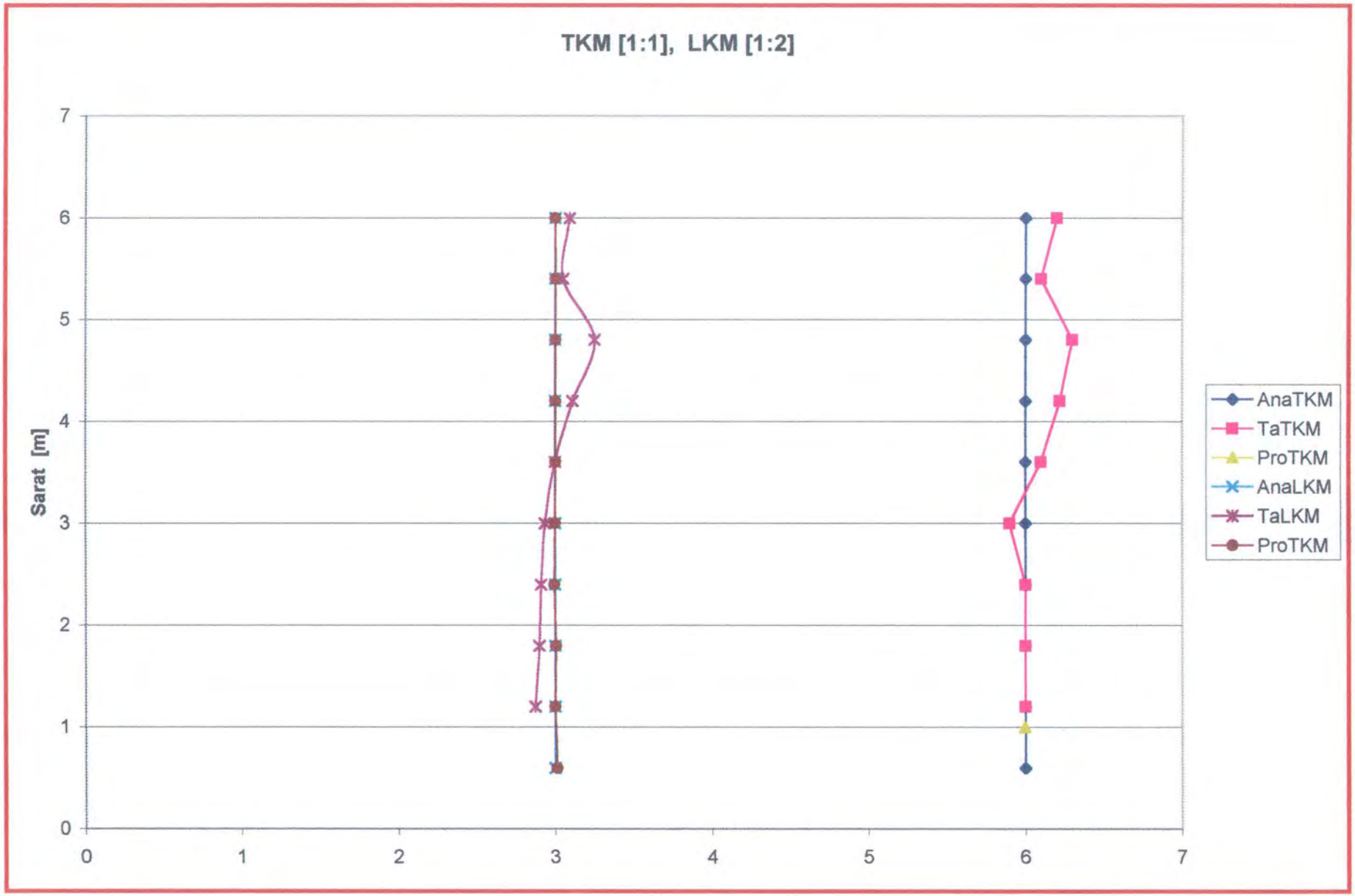

WPA [1:100] m2; Cw[1:1]
Sarat [m]
0
1
2
3
4
5
6
7
0
0.2
0.4
0.6
0.8
1
1.2
AnaWPA
TaWPA
ProWPA
AnaCw
TaCw
ProCw


Displ [1: 400] ton, Cb [1:1]
Sarat [m]
0
1
2
3
4
5
6
7
0
0.2
0.4
0.6
0.8
1
1.2
1.4
AnaDIS
TaDIS
ProDIS
AnaCb
TaCb
ProCb


KB [1:10], Cp [1:1]
Sarat [m]
0
1
2
3
4
5
6
7
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
AnaKB
TaKB
ProKB
AnaCp
TaCp
ProCp


TBM [1:1] , LBM [1:2]
Sarat [m]
0
1
2
3
4
5
6
7
AnaTBM
TaTBM
ProTBM
AnaLBM
TaLBM
ProLBM


WSA [1:400], Cm [1:1}
Sarat [m]
0
1
2
3
4
5
6
7
0
0.2
0.4
0.6
0.8
1
1.2
1.4
AnaWSA
TaWSA
Program
AnaCm
TaCm
ProCm


TKM [1:1], LKM [1:2]
Sarat [m]
0
1
2
3
4
5
6
7
AnaTKM
TaTKM
ProTKM
AnaLKM
TaLKM
ProTKM

DAFTAR PUSTAKA

1. Rogers, David F. and Adams, J. Alan , **Mathematical Elements for Computer Graphics,** *McGraw-Hill,* New York, February 1989.
2. Baihaqqi, Mohammad, **Pendefinisian Persamaan Permukaan Badan Kapal dari Data Badan Kapal Dengan Metode B-Spline Menggunakan PC**, *Jurusan Teknik Perkapalan ITS,* Surabaya, 1995.
3. Nowacky, H., **Ship Lines Creation by Computer, Objectives, Methods and Results**, *Proceedings SCAHSD,* Anapolis, 1977
4. Hearn, Donald and Baker, M. Pauline, **Computer Graphics,** *Prentice-Hall,inc.* , New Jersey
5. Wozniewicz, Andrew J. , Shammas, Namir and Campbell, Tom, **Tech Your Self Delphi in 21 Days,** *Sam Publishing*,Indianapolis, 1995
6. Hill, Francis S., **Computer Graphics,** Macmilian Publishing, New York
7. Okianto, D, **Borland Delphi 2.0 for Windows 95,** *Elex Media Komputindo, Gramedia,* 1996.
8. Calvert, Charles, **Delphi 2 Unleashed,** *Sam Publishing,* Indianapolis, 1996

```
: mother;

erface

s
indows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
enus,IODATA, ExtCtrls,bsplfit, StdCtrls, DB, DBTables,Ta_Decl, Grids, DBGrids,
data3d, calcpoly, calcsurf;


e
Main = class(TForm)
  MainMenu1: TMainMenu;
 File1: TMenuItem;
 Open1: TMenuItem;
 Save1: TMenuItem;
 SaveAs1: TMenuItem;
 Exit1: TMenuItem;
 View1: TMenuItem;
 Calculate1: TMenuItem;
 New1: TMenuItem;
 Label1: TLabel;
 Labeltabel: TLabel;
 Query1: TQuery;
 Query2: TQuery;
 DataSource1: TDataSource;
 OpenDialog1: TOpenDialog;
 SaveDialog1: TSaveDialog;
 DBGrid1: TDBGrid;
 table1: TTable;
 ComboBox1: TComboBox;
 edit1: TEdit;
 Table2: TTable;
 N1: TMenuItem;
 N2: TMenuItem;
 Bevel1: TBevel;
 Label2: TLabel;
 Edit2: TEdit;
 ComboBox2: TComboBox;
 Bevel2: TBevel;
 Bevel3: TBevel;
 RadioButton1: TRadioButton;
 RadioButton2: TRadioButton;
 RadioButton3: TRadioButton;
 RadioButton4: TRadioButton;
 RadioButton5: TRadioButton;
 DataSource2: TDataSource;
 DBGrid2: TDBGrid;
 DataSource3: TDataSource;
 Table3: TTable;
 DBGrid3: TDBGrid;
 Table4: TTable;
 RadioButton6: TRadioButton;
 Datain3DSpace1: TMenuItem;
 Polyfromdata: TMenuItem;
 N3: TMenuItem;
 SurfacefromPolygonnet1: TMenuItem;
 LinesData1: TMenuItem;
 N4: TMenuItem;
 Hidrostatic1: TMenuItem;
 procedure FormCreate(Sender: TObject);
 procedure FormDestroy(Sender: TObject);
 procedure New1Click(Sender: TObject);
 procedure Exit1Click(Sender: TObject);
 procedure Open1Click(Sender: TObject);
```

```
 procedure ComboBox1Change(Sender: TObject);
 procedure table1AfterOpen(DataSet: TDataSet);
 procedure table1AfterClose(DataSet: TDataSet);
 procedure Save1Click(Sender: TObject);
 procedure SaveAs1Click(Sender: TObject);
 procedure ComboBox2Change(Sender: TObject);
 procedure RadioButton1Click(Sender: TObject);
 procedure RadioButton2Click(Sender: TObject);
 procedure RadioButton5Click(Sender: TObject);
 procedure RadioButton3Click(Sender: TObject);
 procedure RadioButton4Click(Sender: TObject);
 procedure RadioButton6Click(Sender: TObject);
 procedure Datain3DSpace1Click(Sender: TObject);
 procedure DBGrid3KeyPress(Sender: TObject; var Key: Char);
 procedure DBGrid1KeyPress(Sender: TObject; var Key: Char);
 procedure DBGrid2KeyPress(Sender: TObject; var Key: Char);
 procedure PolyfromdataClick(Sender: TObject);
 procedure SurfacefromPolygonnet1Click(Sender: TObject);
 procedure LinesData1Click(Sender: TObject);
 procedure FormActivate(Sender: TObject);
 procedure Hidrostatic1Click(Sender: TObject);
private
 Procedure Bukatabel(S:string);
 procedure buattabel(s:string);
 procedure savetabel;
 procedure saveastabel;
 procedure dialogsave;
 procedure pilihan;
 procedure savetobaidata;
  { Private declarations }
ublic
  { Public declarations }
nd;

:
Main: TMain;

plementation

s Chidro;
{ *.DFM}
ocedure TMain.FormCreate(Sender: TObject);
gin
berubah:=false;save1.enabled:=false;
Saveas1.enabled:=false;
Datain3DSpace1.enabled:=false;
PolyfromData.enabled:=false;
file2:=createfile2name(filekerja);
file3:=createfile3name(filekerja);
buattabel(filekerja);
d;

ocedure TMain.FormDestroy(Sender: TObject);
gin
table1.close;table2.close;table3.close;
d;

ocedure TMain.New1Click(Sender: TObject);
 i:integer;
gin
Dialogsave;
f keluar then
egin
  table1.close;table2.close;table3.Close;
  combobox1.text:='';combobox2.text:='';
```

```
  berubah:=false;
  edit1.visible:=false;
  bevel2.visible:=true;
  combobox1.visible:=true;
  edit2.visible:=false;
  combobox2.visible:=true;
  for i:=1 to 100 do
  begin
    combobox1.items.add(inttostr(i));
  end;
  label1.visible:=true;
  for i:=1 to 100 do
  begin
    combobox2.items.add(inttostr(i));
  end;
  label2.visible:=true;
  Main.caption:='Hull Surface Fairing - Untitled.hbd';
nd;
l;

cedure TMain.Exit1Click(Sender: TObject);
in
ialogsave;
f keluar then
egin
  close;
nd;
l;

cedure tMain.dialogsave;
 istr:string;
in
eluar:=true;
f (table1.active)or(table2.active)or(table3.active) then
egin
  if (berubah) then
  begin
    istr:=format('Save changes to "%s" ?',[namafile]);
    case application.messagebox(
      Pchar(istr),
      'Confirm',MB_YESNOCANCEL or MB_ICONHAND) of

       IDYES : begin
                 savetabel;
               end;
       IDCancel:keluar:=false;
       IDNO : keluar:=true;
    end;
  end;
nd;
l;

cedure TMain.buattabel(s:string);
 i:integer;
  istr : string;
in
uery2.close;
Query2.SQL.Clear;
Query2.SQL.Add(Format('CREATE TABLE "%s"',[s]));
Query2.SQL.Add('(');
Query2.SQL.Add('Station integer,');
Query2.SQL.Add('Distance float(8,4),');
for i:=1 to jlh_wl do
egin
  istr:='';
```

```
 istr:='WL_'+inttostr(i)+' float(8,4),';
 Query2.SQL.Add(istr);
nd;
uery2.SQL.Add('Sheer float(8,4),');
uery2.SQL.Add('sheer_D float(8,4),');
uery2.SQL.Add('Keel float(8,4)');
uery2.SQL.Add(')');
uery2.ExecSQL;
uery2.CLose;
uery2.SQL.Clear;
uery2.SQL.Add(Format('CREATE TABLE "%s"',[file2]));
uery2.SQL.Add('(');
uery2.SQL.Add('WL integer,');
uery2.SQL.Add('Distance2 float(8,4),');
uery2.SQL.Add('Stem float(8,4),');
uery2.SQL.Add('stern float(8,4)');
uery2.SQL.Add(')');
uery2.ExecSQL;
uery2.CLose;
uery2.SQL.Clear;
uery2.SQL.Add(Format('CREATE TABLE "%s"',[file3]));
uery2.SQL.Add('(');
uery2.SQL.Add('Stem_Stern CHARACTER (12),');
uery2.SQL.Add('AP_LastSta float(8,4),');
uery2.SQL.Add('Base_Line float(8,4)');
uery2.SQL.Add(')');
uery2.ExecSQL;
uery2.CLose;
l;


ocedure tMain.pilihan;
gin
evel3.visible:=true;
adiobutton1.visible:=true;
adiobutton2.visible:=true;
adiobutton3.visible:=true;
adiobutton4.visible:=true;
adiobutton5.visible:=true;
adiobutton6.visible:=true;
d;

ocedure TMain.Bukatabel(S:string);
r i:integer;
gin
dbgrid1.visible:=false;dbgrid2.visible:=false;dbgrid3.visible:=false;
datasource1.dataset:=query1;
Datasource1.Enabled:=false;
With Query1 do
egin
  Close;
  SQL.Clear;
  SQL.Add(Format('SELECT * FROM "%s"',[s]));
  Open;
  jlh_WL:=fieldcount-5;
  if (Fields[0].FieldName<>'Station')or
     (Fields[1].FieldName<>'Distance') Then
  begin
    Application.Messagebox(
      'File data tidak kompatibel',
      'Informasi',MB_OK or MB_Iconinformation);
    close;
    Namafile:='';
  end;
end;
```

```
atasource1.Enabled:=true;
uery1.close;
able1.tablename:=s;
atasource1.dataset:=table1;
bgrid1.DataSource:=datasource1;
able1.active:=true;
f namafile=filenew then
egin
  for i:=1 to jlh_sta do
  begin
    table1.Append;
    table1.fields[0].value:=i;
  end;
  table1.first;
nd
lse begin
  jlh_sta:=0;
  while not(table1.EOF) do
  begin
    jlh_sta:=jlh_Sta+1;
    table1.next;
  end;
able1.first;
nd;
for i:=0 to table1.fieldcount-1 do
egin
  dbgrid1.columns.Items[i].title.alignment:=tacenter;
  if i=0 then dbgrid1.columns.items[i].alignment:=tacenter
  else (table1.fields[i] as Tfloatfield).Displayformat:='#.0000';
nd;
Datasource2.dataset:=query1;
Datasource2.Enabled:=false;
With Query1 do
egin
  Close;
  SQL.Clear;
  SQL.Add(Format('SELECT * FROM "%s"',[file2]));
  Open;
  if (Fields[0].FieldName<>'WL')or
     (Fields[1].FieldName<>'Distance2') Then
  begin
    Application.Messagebox(
      'File data tidak kompatibel',
      'Informasi',MB_OK or MB_Iconinformation);
    close;
    Namafile:='';
  end;
end;
Datasource2.Enabled:=true;
query1.close;
table2.tablename:=file2;
datasource2.dataset:=table2;
dbgrid2.DataSource:=datasource2;
table2.active:=true;
for i:=0 to table2.fieldcount-1 do
begin
  dbgrid2.columns.Items[i].title.alignment:=tacenter;
  if i=0 then dbgrid2.columns.items[i].alignment:=tacenter
  else begin
    dbgrid2.fields[i].displaywidth:=30;
   (table2.fields[i] as Tfloatfield).Displayformat:='#.0000';
  end;
end;
if namafile=filenew then
begin
```

```
 for i:=1 to jlh_WL do
 begin
   table2.Append;
   table2.fields[0].value:=inttostr(i);
 end;
 table2.first;
ad;
atasource3.dataset:=query1;
atasource3.Enabled:=false;
ith Query1 do
egin
 Close;
 SQL.Clear;
 SQL.Add(Format('SELECT * FROM "%s"',[file3]));
 Open;
 if (Fields[0].FieldName<>'Stem_Stern')or
    (Fields[1].FieldName<>'AP_LastSta') Then
 begin
   Application.Messagebox(
     'File data tidak kompatibel',
     'Informasi',MB_OK or MB_Iconinformation);
   close;
   Namafile:='';
 end;
nd;
atasource3.Enabled:=true;
uery1.close;
able3.tablename:=file3;
atasource3.dataset:=table3;
bgrid3.DataSource:=datasource3;
able3.active:=true;
or i:=0 to table3.fieldcount-1 do
egin
 dbgrid3.columns.Items[i].title.alignment:=tacenter;
 if i<>0 then
 begin
  (table3.fields[i] as Tfloatfield).Displayformat:='#.0000';
   dbgrid3.Fields[i].displaywidth:=30;
 end
 else dbgrid2.columns.items[i].alignment:=tacenter;
nd;
f namafile=filenew then
egin
 table3.Append;
 table3.fields[0].value:='Stem';
 table3.Append;
 table3.fields[0].value:='Stern';
 table3.first;
nd;
;

ocedure TMain.ComboBox1Change(Sender: TObject);
in
f (combobox1.text<>'')and(combobox2.text<>'') then
egin
 edit1.text:=combobox1.text;
 edit1.visible:=true;
 combobox1.visible:=false;
 edit2.text:=combobox2.text;
 edit2.visible:=true;
 combobox2.visible:=false;
 jlh_sta:=strtoint(edit2.text);
 jlh_WL:=strtoint(edit1.text);
 pilihan;
 file2:=createfile2name(filekerja);
```

```
 file3:=createfile3name(filekerja);
 buattabel(filekerja);
 namafile:=filenew;
 BukaTabel(Filekerja);
 radiobutton1.checked:=true;radiobutton1.OnClick(sender);
nd;
;

cedure TMain.ComboBox2Change(Sender: TObject);
in
f (combobox1.text<>'')and(combobox2.text<>'') then
egin
 edit1.text:=combobox1.text;
 edit1.visible:=true;
 combobox1.visible:=false;
 edit2.text:=combobox2.text;
 edit2.visible:=true;
 combobox2.visible:=false;
 jlh_sta:=strtoint(edit2.text);
 jlh_WL:=strtoint(edit1.text);
 pilihan;
 file2:=createfile2name(filekerja);
 file3:=createfile3name(filekerja);
 buattabel(filekerja);
 namafile:=filenew;
 BukaTabel(Filekerja);
 radiobutton1.checked:=true;radiobutton1.OnClick(sender);
nd;
l;

ocedure TMain.RadioButton1Click(Sender: TObject);
: i:integer;
gin
abeltabel.visible:=true;
abeltabel.caption:='Half Breadth at each Water Lines and Sheer Lines';
ith table1 do
egin
or i:=0 to fieldcount-1 do
egin
 if (i=1)or(i=fieldcount-1)or(i=fieldcount-2) then
 begin
   if (fields[i].visible)then fields[i].visible:=false;
 end
 else if (fields[i].visible=false)then fields[i].visible:=true;
nd;
nd;
f Dbgrid2.visible=true then Dbgrid2.visible:=false;
f Dbgrid3.visible=true then Dbgrid3.visible:=false;
f (Dbgrid1.visible=false)then Dbgrid1.visible:=true;
;

ocedure TMain.RadioButton2Click(Sender: TObject);
: i:integer;
gin
abeltabel.visible:=true;
abeltabel.caption:='Distance each Station from AP';
ith table1 do
egin
or i:=0 to fieldcount-1 do
egin
 if (i=0)or(i=1) then
 begin
   if (fields[i].visible=false)then fields[i].visible:=true;
 end
 else if (fields[i].visible)then fields[i].visible:=false;
```

```
nd;
nd;
f Dbgrid2.visible=true then Dbgrid2.visible:=false;
f Dbgrid3.visible=true then Dbgrid3.visible:=false;
f (Dbgrid1.visible=false)then Dbgrid1.visible:=true;
;

cedure TMain.RadioButton5Click(Sender: TObject);
 i:integer;
in
abeltabel.visible:=true;
abeltabel.caption:='Distance Sheer Lines from Base Line at each Station';
ith table1 do
egin
or i:=0 to fieldcount-1 do
egin
 if (i=0)or(i=fieldcount-2) then
 begin
   if (fields[i].visible=false)then fields[i].visible:=true;
 end
 else if (fields[i].visible)then fields[i].visible:=false;
nd;
ields[fieldcount-2].displaylabel:='Distance';
nd;
f Dbgrid2.visible=true then Dbgrid2.visible:=false;
f Dbgrid3.visible=true then Dbgrid3.visible:=false;
f (Dbgrid1.visible=false)then Dbgrid1.visible:=true;
l;

ocedure TMain.RadioButton3Click(Sender: TObject);
: i:integer;
in
beltabel.visible:=true;
beltabel.caption:='Distance each Water Lines from Base Line, Stem and Stern';
th table2 do
gin
or i:=0 to fieldcount-1 do
egin
   if (fields[i].visible=false)then fields[i].visible:=true;
nd;
ields[1].displaylabel:='Distance WL from Base Line';
ields[2].displaylabel:='Distance stem from station '+inttostr(jlh_sta);
ields[3].displaylabel:='Distance stern from AP';
d;
 Dbgrid1.visible=true then Dbgrid1.visible:=false;
 Dbgrid3.visible=true then Dbgrid3.visible:=false;
 (Dbgrid2.visible=false)then Dbgrid2.visible:=true;
;

ocedure TMain.RadioButton4Click(Sender: TObject);
in
abeltabel.visible:=true;
abeltabel.caption:='Distance stem stern at sheer from AP and Base Line';
bgrid3.fields[1].displaylabel:='Distance from AP';
bgrid3.fields[2].displaylabel:='Distance from base line';
f Dbgrid1.visible then Dbgrid1.visible:=false;
f Dbgrid2.visible then Dbgrid2.visible:=false;
f (Dbgrid3.visible=false)then Dbgrid3.visible:=true;
;

ocedure TMain.RadioButton6Click(Sender: TObject);
 i:integer;
in
abeltabel.visible:=true;
abeltabel.caption:='Distance Keel from Base Line at each Station';
```

```
ith table1 do
egin
or i:=0 to fieldcount-1 do
egin
 if (i=0)or(i=fieldcount-1) then
 begin
   if (fields[i].visible=false)then fields[i].visible:=true;
 end
 else if (fields[i].visible)then fields[i].visible:=false;
nd;
ields[fieldcount-1].displaylabel:='Distance';
nd;
f Dbgrid2.visible=true then Dbgrid2.visible:=false;
f Dbgrid3.visible=true then Dbgrid3.visible:=false;
f (Dbgrid1.visible=false)then Dbgrid1.visible:=true;
;

cedure TMain.Savetabel;
in
eluar:=true;
f namafile=filenew then
egin
 saveastabel;
nd
lse begin
 berubah:=false;
 file2:=createfile2name(namafile);
 file3:=createfile3name(namafile);
 buattabel(namafile);
 table4.tablename:=namafile;
 table4.active:=true;
 Table4.BatchMove(Table1, batAppend);
 table4.close;
 table4.tablename:=file2;
 table4.active:=true;
 Table4.BatchMove(Table2, batAppend);
 table4.close;
 table4.tablename:=file3;
 table4.active:=true;
 Table4.BatchMove(Table3, batAppend);
 table4.close;
nd;
d;


ocedure TMain.Open1Click(Sender: TObject);
 i:integer;
gin
Dialogsave;
pendialog1.filter:='Half breadth data (*.Hbd)|*.hbd';
if keluar then
egin
 if (OpenDialog1.Execute) then
 begin
   berubah:=false;
   table1.close;table2.close;table3.close;
   pilihan;
   namafile:=opendialog1.FileName;
   namacaption:='';
   for i:=1 to length(namafile) do
   begin
     if namafile[i]='\' then namacaption:=''
     else namacaption:=namacaption+namafile[i];
   end;
   Main.caption:='Hull Surface Hidrostatic - '+namacaption;
```

```
  file2:=createfile2name(opendialog1.filename);
  file3:=createfile3name(opendialog1.filename);
  Bukatabel(OpenDialog1.Filename);
  namafile:=filekerja;
  savetabel;
  table1.close;table2.close;table3.close;
  bukatabel(filekerja);
  bevel2.visible:=true;
  label1.visible:=true;label2.visible:=true;
  edit1.visible:=true;edit2.visible:=true;
  edit1.text:=inttostr(jlh_WL);
  edit2.text:=inttostr(jlh_sta);
  namafile:=opendialog1.filename;
  radiobutton1.checked:=true;radiobutton1.OnClick(sender);
end;
nd;
;

cedure TMain.table1AfterOpen(DataSet: TDataSet);
in
ave1.enabled:=true;
aveas1.enabled:=true;
atain3DSpace1.enabled:=true;
olyfromData.enabled:=true;
;

cedure TMain.table1AfterClose(DataSet: TDataSet);
in
ave1.enabled:=false;
aveas1.enabled:=false;
;

cedure TMain.Save1Click(Sender: TObject);
in
savetabel;
;


cedure TMain.saveastabel;
 i:integer;
in
avedialog1.filter:='Half Breadth Data (*.hbd)|*.hbd';
f savedialog1.execute then
egin
 namafile:=savedialog1.filename;
 if namafile[length(namafile)-3]<>'.' then
 namafile:=namafile+'.hbd';
 namacaption:='';
 for i:=1 to length(namafile) do
 begin
   if namafile[i]='\' then namacaption:=''
   else namacaption:=namacaption+namafile[i];
 end;
 Main.caption:='Hull Surface Hidrostatic - '+namacaption;
 file2:=createfile2name(namafile);
 file3:=createfile3name(namafile);
 buattabel(namafile);
 table4.tablename:=namafile;
 table4.active:=true;
 Table4.BatchMove(Table1, batAppend);
 table4.close;
 table4.tablename:=file2;
 table4.active:=true;
 Table4.BatchMove(Table2, batAppend);
 table4.close;
```

```
  table4.tablename:=file3;
  table4.active:=true;
  Table4.BatchMove(Table3, batAppend);
  table4.close;
  keluar:=true;
nd
lse keluar:=false;
;

cedure TMain.SaveAs1Click(Sender: TObject);
in
aveastabel;
;

cedure TMain.DBGrid3KeyPress(Sender: TObject; var Key: Char);
in
berubah:=true;
;

cedure TMain.DBGrid1KeyPress(Sender: TObject; var Key: Char);
in
berubah:=true;
;

cedure TMain.DBGrid2KeyPress(Sender: TObject; var Key: Char);
in
erubah:=true;
;

cedure TMain.Datain3DSpace1Click(Sender: TObject);
in
savetobaidata;
jlh_stabef:=jlh_sta;
jlh_WLbef:=jlh_WL;
pendat(baifile,tempd);
showwhat:=1{HB};
ertama:=true;
formview.showmodal;
jlh_sta:=jlh_stabef;
jlh_WL:=jlh_WLbef;
;


cedure TMain.Polyfromdata1Click(Sender: TObject);
in
savetobaidata;
calcplan:='WL';
openfirsttime:=true;
jlh_stabef:=jlh_sta;
jlh_WLbef:=jlh_WL;
formpoly.timer1.enabled:=true;
formpoly.showmodal;
jlh_sta:=jlh_stabef;
jlh_WL:=jlh_WLbef;
;


cedure TMain.SurfacefromPolygonnet1Click(Sender: TObject);
 i,j:integer;
in
opendialog1.filter:='poligon data (*.plg)|*.plg';
if opendialog1.execute then
begin
  jlh_stabef:=jlh_sta;
  jlh_WLbef:=jlh_WL;
```

```
 polyfilename:=Opendialog1.filename;
 for i:=1 to 3  do new(b[i]);
 openpoly(okp);
 if okp then
 begin
   fairingpoly;
   getsignpoly;
   for i:=1 to jlh_sta*jlh_WL do
   begin
     for j:=1 to 3 do
     tempD[j,i]:=b[j]^[i];
   end;
   bspline;
   formcalcsurf.showmodal;
 end
 else begin
   application.messagebox('Polygon File Error',
    'Confirm',MB_OK or MB_ICONHAND);
 end;
 jlh_sta:=jlh_stabef;
 jlh_WL:=jlh_WLbef;
 for i:=1 to 3  do dispose(b[i]);
nd;
l;

cedure TMain.savetobaidata;
:
,j,ii :word;
asukan:string;
in
able1.disablecontrols;
f not(table1.bof) then table1.first;
sh:=1;wk:=1;
or i:=1 to jlh_sta do
egin
  if table1.Fields[jlh_wl+2].value=null then wsh:=0;
  if table1.fields[jlh_wl+4].value=null then wk:=0;
  if not(table1.EOF) then table1.next;
nd;
aifile:=namafile;
aifile[length(baifile)]:='m';
ssignfile(fl,baifile);
{$I-} rewrite(fl);{$I+}
f ioresult <>0 then exit;
riteln(fl,'¥');
riteln(fl,'1');
riteln(fl,jlh_sta,'  ',jlh_wl);
riteln(fl,wsh,'  ',wk);
rite(fl,'0.0000  ');
or i:=1 to jlh_wl do write(fl,'1.0000  ');
riteln(fl);
rite(fl,'0.0000  ');
able2.disablecontrols;
f not(table2.bof) then table2.first;
or j:=1 to jlh_WL do
egin
  masukan:=formatfloat('0.0000',table2.fields[1].value);
  for ii:=1 to length(masukan)do
  if masukan[ii]=','then masukan[ii]:='.';
  write(fl,masukan,'  ');
  if not(table2.EOF) then table2.next;
nd;
f not(table1.bof) then table1.First;
hipbreadth:=0;
or i:=1 to jlh_sta do
```

```
egin
  writeln(fl);
  masukan:=formatfloat('0.0000',table1.fields[1].value);
  for ii:=1 to length(masukan)do
  if masukan[ii]=','then masukan[ii]:='.';
  write(fl,masukan,'  ');
  for j:=1 to jlh_wl do
  begin
    if shipbreadth<table1.fields[j+1].value then
    shipbreadth:=table1.fields[j+1].value;
    masukan:=formatfloat('0.0000',table1.fields[j+1].value);
    for ii:=1 to length(masukan)do
    if masukan[ii]=','then masukan[ii]:='.';
    write(fl,masukan,'  ');
  end;
  if not(table1.EOF) then table1.Next;
nd;
f not(table2.bof) then table2.first;
or i:=1 to jlh_wl do
egin
  writeln(fl);
  masukan:=formatfloat('0.00000',table2.fields[3].value);
  for ii:=1 to length(masukan)do
  if masukan[ii]=','then masukan[ii]:='.';
  write(fl,masukan,'  ');
  masukan:=formatfloat('0.00000',table2.fields[2].value);
  for ii:=1 to length(masukan)do
  if masukan[ii]=','then masukan[ii]:='.';
  write(fl,masukan,'  ');
  if not(table2.EOF) then table2.next;
nd;

 if wsh=1 then
egin
  writeln(fl);
  write(fl,ssz[1],'  ',ssz[2]);
  for i:=1 to 2 do
  begin
    writeln(fl);
    for j:=1 to nstat do write(fl,sheer[i]^[j],'  ');
  end;
nd; }
 if wk=1 then
or j:=1 to nstat do write(fl,keel^[j],'  ');

losefile(fl);
able1.enablecontrols;table2.enablecontrols;
d;


ocedure TMain.LinesData1Click(Sender: TObject);
r
:integer;
k:boolean;
gin
pendialog1.filter:='Lines data (*.Lns)|*.Lns';
f Opendialog1.execute then
egin
  jlh_stabef:=jlh_sta;
  jlh_WLbef:=jlh_WL;
  filelines:=opendialog1.filename;
  for i:=1 to 3 do new(bdcor[i]);
  for i:=1 to 3 do new(wlcor[i]);
  for i:=1 to 3 do new(blcor[i]);
  for i:=1 to 3 do new(sscor[i]);
```

```
  new(gvcor);
  for i:=1 to 3 do new(b[i]);
  Openline(OK);
  if OK then openpoly(OK);
  if OK then
  begin
    getsignpoly;
    fairingpoly;
    showwhat:=4{LINES};
    pertama:=true;showgauss:=true;
    formview.showmodal;
  end;
  jlh_sta:=jlh_stabef;
  jlh_WL:=jlh_WLbef;
  for i:=1 to 3 do  dispose(bdcor[i]);
  for i:=1 to 3 do  dispose(wlcor[i]);
  for i:=1 to 3 do  dispose(blcor[i]);
  for i:=1 to 3 do  dispose(sscor[i]);
  dispose(gvcor);
  for i:=1 to 3 do  dispose(b[i]);
nd;
l;

ocedure TMain.FormActivate(Sender: TObject);
gin
eight:=ClientHeight;
Width:=ClientWidth;
Windowstate:=WsMaximized;
l;

ocedure TMain.Hidrostatic1Click(Sender: TObject);
r i:integer;
gin
pendialog1.filter:='Half Breadth for Hidrostatic (*.HBH)|*.HBH';
if (OpenDialog1.Execute) then
egin
  for i:=1 to 3 do new(bdcor[i]);
  fileHBfB:=opendialog1.filename;
  openHBfB;
  FormHidrstk.showmodal;
  for i:=1 to 3 do dispose(bdcor[i]);
nd;
l;

l.
```

```
;

it Ta_Decl;
terface
nction Pangkat(var A :real;B : Real):Double;
nction createfile2name(nam:String):String;
nction createfile3name(nam:String):String;
nction CreatepolyFilename(nam:String):String;
nction stringisinteger(s:string):boolean;
nction floatisinteger(s:string):boolean;

nst

Filenew='c:\Darwin1\datadelp\untitled.hbd';
Filekerja='c:\Darwin1\datadelp\fucker.wrk';
pointfilename='c:\Darwin1\datadelp\point.pts';
gaussfilename='c:\Darwin1\datadelp\gauss.gcv';
FairPolyfile='c:\Darwin1\datadelp\fairp.pol';
FairPolyfile2='c:\Darwin1\datadelp\fairp.po2';
gnum=150;
gprt=50;
rom ctv}
 maxdpts=11000;
 maxpts=11000;
 max3=maxdpts+3;
 maxstat=250;
 maxwl=200;
 maxque=50;

pe
romctv}
 dptstype=array[1..maxdpts]of single;
 pointtyp3=array[1..3,1..maxdpts] of single;
 pointtype=array[1..3] of ^dptstype;
 mattype  = array[1..1,1..maxstat] of single;
 knottype = array[1..maxstat] of single;
 ptstype  = array[1..maxpts] of single;
 plottype = array[1..3] of ^ptstype;
 dat2type = array[1..2] of ^dptstype;
 fit3type = array[1..max3] of single;
 wltype   = array[0..maxwl] of single;
 stattype = array[1..500] of single;
 polytt   = array[1..3] of ^stattype;
 plotpar  = record
              pl,par : single;
            end;
 gausstype = record
               paru,parw,x,y,z,warna:single;
             end;
 gausstypearray=array[1..maxdpts]of gausstype;
 gausspointtype=^gausstypearray;
 pptype   = array[1..maxpts] of plotpar;
 p2dtype  = array[1..500] of plotpar;
 puwtype  = array[1..3] of ^pptype;
 xyztype  = array[1..3] of single;
 xy       = array[1..2] of single;
 four     = array[1..4] of word;
 gaussfiletype=file of gausstype;
ar
bsignx,bsignz:array[1..maxdpts]of integer;
MyKey:Char;
tilt:single;
increm:real=0.0100;
bawal:single;
tempD: pointtyp3;
gvcor:gausspointtype;
gaussminneg,gaussmaxneg,gaussminpos,gaussmaxpos,
```

```
entangwarnapos,rentangwarnaneg,shipbreadth :single;
o1:integer=10;
o2:integer=25;
hangepolpo2,viewgausscurv,viewpolyvert,selectcondition,
hangepolystatus,changepolyada:boolean;
otaliterasi,Nopolychange:integer;
arnaf:array [0..11] of longint;
olor_use:longint;
aussbts:array[1..14]of single;
arna:array[1..11000] of longint;
um:integer;
allpaint,updateview:boolean;
aussfilename2:string;
aussfile,gaussfile2:gaussfiletype;
aussvar  : gausstype;
sec,zsec,loa:single;
lsect,wlsect,bdsect  : array[1..maxstat] of single;
alcplan :string[15];
howwhat,lmn:word;
latafile:string;
rderk:word=3;
rderl:word=3;
l:textfile;
,y              : knottype;
lh_sta,Jlh_stabef,jlh_WLbef,jlh_wl,Upolno,Wpolno  :integer;
lata2file,polyfilename,namacaption,namafile,file2,file3,
baifile,baifile2,baifile3,filelines,fileHBfB :string;
nbasis,mbasis,
md1basis,md2basis,nd1basis,nd2basis: mattype;
pertama,pertama2,keluar,berubah,okp,openfirsttime,showfirsttime,
progresson,gaussshowfirst,showgauss :boolean;
mode                                :word;
ViewActive,STactive,WLActive,WLActivebef,STActiveBef,STactiveST,WLactiveWL:integer;
*************************************)
{from ctv}
bsc,dnew,b2d,b2dnew,blcor,
bdcor,wlcor,b       :pointtype;
bknot           : dat2type;
eps             : ^dptstype;
P               : puwtype;
sc,sc2          : plottype;
p2d,p2dsc       : array[1..2] of ^p2dtype;
t,u,w,xcen,zcen : single;
wl,maxb         : ^wltype;
stat,partemp,keel: ^stattype;
upar,wpar       : ^ptstype;
polytemp,sscor  : polytt;
newwl,
hmg,newstat     : stattype;
user,smallf,
wsh,wk          : byte;
title           : string;
ssz             : array[1..2] of single;
xwl             : array[1..maxstat] of ^knottype;
yst             : array[1..maxstat] of ^knottype;
ssx             : array[1..2,1..maxwl] of single;
coords          : array[1..3,1..maxstat] of single;
nstatss         : array[1..maxstat] of word;
lwl             : array[1..maxwl]of single;
p1,p2,p3    : xyztype;
sheer           : array[1..2] of ^stattype;
showdat,center,Lcenter,showbd,showwl,showbl,
HBcenter,full,withbot,showerr  : boolean;

{from ctv}
```

```
 blnum,wlnum,bdnum,blprt,bdprt,wlprt,x0,y0,z0,i,nplusc,mplusc,chst,
 poBP,nwlmod,nstatmod,row,shin,rendnum,mx,my,modedata,igen,
 fsec,x1,x2,y1,y2 : integer;

 scale,shift,diss,vds,ss,thetas,phis,cxs,cys,vxs,vys,Ldis,kd,ks,Ia,Ip,d0,
 wlormax,stormax,asx,asz,dis,vd,s,scf,vx,Lthet,Lphi,Lsn1,Lsn2,Lcs1,Lcs2,
 vy,theta0,phi0,cx,cy,cn1,cn2,sn1,sn2,theta,phi: single;


plementation

nction Pangkat(var A:real;B : Real):double;
gin
  pangkat := Exp(Ln(A)*B);
d;

nction CreateFile2name(nam:String):String;
r
namaFiler : String;
gin
namaFiler := '';
namaFiler := nam;
namafiler[length(nam)]:='2';
CreateFile2Name := namaFiler;
d;

nction CreateFile3name(nam:String):String;
r
namaFiler : String;
gin
namaFiler := '';
namaFiler := nam;
namafiler[length(nam)]:='3';
CreateFile3Name := namaFiler;
d;

nction CreatepolyFilename(nam:String):String;
r
namaFiler : String;
gin
namaFiler := '';
namaFiler := nam;
namafiler[length(nam)]:='g';
namafiler[length(nam)-1]:='l';
namafiler[length(nam)-2]:='p';
CreatepolyFileName := namaFiler;
d;

nction stringisinteger(s:string):boolean;
r i:integer;
  benar:boolean;
gin
benar:=true;
for i:=1 to length(s) do
begin
  if not(s[i]in['-','1','2','3','4','5','6','7','8','9','0']) then
  benar:=false;
end;
result:=benar;
d;


nction floatisinteger(s:string):boolean;
r i,j:integer;
```

```
 benar:boolean;
in
:=0;
enar:=true;
or i:=1 to length(s) do
egin
 if not(s[i]in['1','2','3','4','5','6','7','8','9','0','.']) then
 benar:=false;
nd;
f benar then
egin
 for i:=1 to length(s) do
 begin
   if (s[i]='.') then j:=j+1;
 end;
nd;
f j>1 then benar:=false;
esult:=benar;
;

.
```

```
it Newton:

terface
es Ta_decl,basknot;
ocedure Searchy(cor1,cor2:single;var corfound:single);
ocedure Searchx(cor1,cor2:single;var corfound:single);
ocedure SearchyBP(mode,nost:word; zcor:single;var found:single);
ocedure SearchyWL(mode,nowl:word; xcor:single;var found:single);
ocedure SearchSS(mode,nst:word;xcor:single; var found:single);
rocedure SearchZ(cor1,cor2:single;var corfound:single);}
ocedure SearchU(mode:word;wpos,ycor:single;var corf1,corf2:single);
ocedure SearchBLE(ycor:single; var found:single);
ocedure Searchsk(mode:word;xcor:single;var corf:single);
ocedure Searchblesk(model,mode2:word;ycor:single;var corf:single);
ocedure SearchxyBP(nost:word; zcor:single;var xfound,yfound:single);
ocedure Searchygauss(cor1,cor2:single;var corfound1,corfound2:single;var
trfound:char);
ocedure SearchSSgauss(mode,nst:word;xcor:single; var found,found2:single;var
trfound:char);


plementation

==========Searching y coordinat==========}
ocedure Searchy(cor1,cor2:single;var corfound:single);
pe
xz = array[1..2] of real;
r
hit,x1,x2,xm,z1,z2,zm,i,j1,j2,j3 : integer;
found : real;
midpl,estpar,estpl,der : xz;

gin
x1:=1;x2:=po1;
z1:=1;z2:=po2;
repeat
  xm:=round((x1+x2)/2);
  zm:=round((z1+z2)/2);
  midpl[1]:=p[1]^[(xm-1)*po2+zm].pl;
  midpl[2]:=p[3]^[(xm-1)*po2+zm].pl;

  if (midpl[1]>=cor1) and (midpl[2]>=cor2) then
  begin
    x2:=xm;z2:=zm;
  end
  else if (midpl[1]>=cor1) and (midpl[2]<cor2) then
  begin
    x2:=xm;z1:=zm;
  end
  else if (midpl[1]<cor1) and (midpl[2]>=cor2) then
  begin
    x1:=xm;z2:=zm;
  end
  else if (midpl[1]<cor1) and (midpl[2]<cor2) then
  begin
    x1:=xm;z1:=zm;
  end;
until ((x2-x1)<=1) and ((z2-z1)<=1);

estpar[1]:=p[1]^[(xm-1)*po2+zm].par;
estpar[2]:=p[3]^[(xm-1)*po2+zm].par;
totaliterasi:=0;
repeat
  totaliterasi:=totaliterasi+1;
```

```
  estpl[1]:=0;estpl[2]:=0;der[1]:=0;der[2]:=0;
  DBasis(orderk,jlh_sta,estpar[1],x,nbasis,nd1basis);
  DBasis(orderl,jlh_WL,estpar[2],y,mbasis,md1basis);
  for j1:=1 to jlh_sta do
    for j2:=1 to jlh_WL do
    begin
      estpl[1]:=estpl[1]+b[1]^[(j1-1)*jlh_WL+j2]*nbasis[1,j1]*mbasis[1,j2];
      estpl[2]:=estpl[2]+b[3]^[(j1-1)*jlh_WL+j2]*nbasis[1,j1]*mbasis[1,j2];
      der[1]:=der[1]+b[1]^[(j1-1)*jlh_WL+j2]*nd1basis[1,j1]*mbasis[1,j2];
      der[2]:=der[2]+b[3]^[(j1-1)*jlh_WL+j2]*nbasis[1,j1]*md1basis[1,j2];
    end;
   writeln(estpl[1]:9:7,'  ',estpl[2]:9:7,'  ',der[1]:9:7,'  ',der[2]:9:7);}

  if abs(estpl[1]-cor1)<1e-4 then estpar[1]:=estpar[1]
  else estpar[1]:=estpar[1]-((estpl[1]-cor1)/der[1]);
  if estpar[1]<0 then estpar[1]:=0
    else if estpar[1]>x[jlh_sta+orderk] then estpar[1]:=x[jlh_sta+orderk];
  if abs(estpl[2]-cor2)<1e-4 then estpar[2]:=estpar[2]
  else estpar[2]:=estpar[2]-((estpl[2]-cor2)/der[2]);
  if estpar[2]<0 then estpar[2]:=0
    else if estpar[2]>y[jlh_WL+orderl] then estpar[2]:=y[jlh_WL+orderl];

until (((abs(estpl[1]-cor1))< 1e-3)and ((abs (estpl[2]-cor2) ) < 1e-3))
      or (totaliterasi>50);
 if totaliterasi>50 then exit;
 totaliterasi:=0;
 found:=0;
 Basis(orderk,jlh_sta,estpar[1],x,nbasis);
 Basis(orderl,jlh_WL,estpar[2],y,mbasis);
 for j1:=1 to jlh_sta do
   for j2:=1 to jlh_WL do
     found:=found+b[2]^[(j1-1)*jlh_WL+j2]*nbasis[1,j1]*mbasis[1,j2];
 corfound:=found;
nd;

rocedure Searchygauss(cor1,cor2:single;var corfound1,corfound2:single;var
strfound:char);
ype
 xz = array[1..2] of real;
ar
 hit,x1,x2,xm,z1,z2,zm,i,j1,j2,j3 : integer;
 found1,found2,AA,BB,CC,QuxQw4    : single;
 Qu,Qw,Quw,Quu,Qww,QuxQw          : array[1..3] of single;
 midpl,estpar,estpl,der           : xz;

egin
 x1:=1;x2:=po1;
 z1:=1;z2:=po2;
 repeat
   xm:=round((x1+x2)/2);
   zm:=round((z1+z2)/2);
   midpl[1]:=p[1]^[(xm-1)*po2+zm].pl;
   midpl[2]:=p[3]^[(xm-1)*po2+zm].pl;
   if (midpl[1]>=cor1)and(midpl[2]>=cor2) then
   begin
     x2:=xm;z2:=zm;
   end
   else if (midpl[1]>=cor1) and (midpl[2]<cor2) then
   begin
     x2:=xm;z1:=zm;
   end
   else if (midpl[1]<cor1) and (midpl[2]>=cor2) then
   begin
     x1:=xm;z2:=zm;
   end
```

```
  else if (midpl[1]<cor1) and (midpl[2]<cor2) then
  begin
    x1:=xm;z1:=zm;
  end;
until ((x2-x1)<=1) and ((z2-z1)<=1);

estpar[1]:=p[1]^[(xm-1)*po2+zm].par;
estpar[2]:=p[3]^[(xm-1)*po2+zm].par;
totaliterasi:=0;
repeat
  totaliterasi:=totaliterasi+1;
  estpl[1]:=0;estpl[2]:=0;der[1]:=0;der[2]:=0;
  DBasis(orderk,jlh_sta,estpar[1],x,nbasis,nd1basis);
  DBasis(orderl,jlh_WL,estpar[2],y,mbasis,md1basis);
  for j1:=1 to jlh_sta do
    for j2:=1 to jlh_WL do
    begin
      estpl[1]:=estpl[1]+b[1]^[(j1-1)*jlh_WL+j2]*nbasis[1,j1]*mbasis[1,j2];
      estpl[2]:=estpl[2]+b[3]^[(j1-1)*jlh_WL+j2]*nbasis[1,j1]*mbasis[1,j2];
      der[1]:=der[1]+b[1]^[(j1-1)*jlh_WL+j2]*nd1basis[1,j1]*mbasis[1,j2];
      der[2]:=der[2]+b[3]^[(j1-1)*jlh_WL+j2]*nbasis[1,j1]*md1basis[1,j2];
    end;
  if abs(estpl[1]-cor1)<1e-4 then estpar[1]:=estpar[1]
  else estpar[1]:=estpar[1]-((estpl[1]-cor1)/der[1]);
  if estpar[1]<0 then estpar[1]:=0
    else if estpar[1]>x[jlh_sta+orderk] then estpar[1]:=x[jlh_sta+orderk];
  if abs(estpl[2]-cor2)<1e-4 then estpar[2]:=estpar[2]
  else estpar[2]:=estpar[2]-((estpl[2]-cor2)/der[2]);
  if estpar[2]<0 then estpar[2]:=0
    else if estpar[2]>y[jlh_WL+orderl] then estpar[2]:=y[jlh_WL+orderl];
until (((abs(estpl[1]-cor1))< 1e-3)and ((abs (estpl[2]-cor2) ) < 1e-3))
      or (totaliterasi>50);
if totaliterasi>50 then exit;
totaliterasi:=0;
strfound:='t';
found1:=0;
found2:=0;
for j1:=1 to 3 do
begin
  qu[j1]:=0;
  qw[j1]:=0;
  quw[j1]:=0;
  quu[j1]:=0;
  qww[j1]:=0;
end;
D2Basis(orderk,jlh_sta,estpar[1],x,nbasis,nd1basis,nd2basis);
D2Basis(orderl,jlh_WL,estpar[2],y,mbasis,md1basis,md2basis);
for j1:=1 to jlh_sta do
for j2:=1 to jlh_WL do
begin
  found1:=found1+b[2]^[(j1-1)*jlh_WL+j2]*nbasis[1,j1]*mbasis[1,j2];
  for j3:=1 to 3 do
  begin
    qu[j3]:=qu[j3]+b[j3]^[(j1-1)*jlh_WL+j2]*nd1basis[1,j1]*mbasis[1,j2];
    qw[j3]:=qw[j3]+b[j3]^[(j1-1)*jlh_WL+j2]*nbasis[1,j1]*md1basis[1,j2];
    quw[j3]:=quw[j3]+b[j3]^[(j1-1)*jlh_WL+j2]*nd1basis[1,j1]*md1basis[1,j2];
    quu[j3]:=quu[j3]+b[j3]^[(j1-1)*jlh_WL+j2]*nd2basis[1,j1]*mbasis[1,j2];
    qww[j3]:=qww[j3]+b[j3]^[(j1-1)*jlh_WL+j2]*nbasis[1,j1]*md2basis[1,j2];
  end;
end;
QuxQw[1]:=qu[2]*qw[3]-qu[3]*qw[2];
QuxQw[2]:=qu[3]*qw[1]-qu[1]*qw[3];
QuxQw[3]:=qu[1]*qw[2]-qu[2]*qw[1];
QuxQw4:=sqr(sqr(QuxQw[1])+sqr(QuxQw[2])+sqr(QuxQw[3]));
AA:=QuxQw[1]*Quu[1]+QuxQw[2]*Quu[2]+QuxQw[3]*Quu[3];
```

```
B:=QuxQw[1]*Quw[1]+QuxQw[2]*Quw[2]+QuxQw[3]*Quw[3];
C:=QuxQw[1]*Qww[1]+QuxQw[2]*Qww[2]+QuxQw[3]*Qww[3];
f QuxQw4<>0 then
egin
 found2:=(AA*CC-sqr(BB))/QuxQw4;
 strfound:='T';
nd
lse
egin
 strfound:='F';found2:=0;
 if (AA*CC-sqr(BB))=0 then strfound:='0';
 if (AA*CC-sqr(BB))<0 then strfound:='-';
 if (AA*CC-sqr(BB))>0 then strfound:='+';
nd;
orfound1:=found1;
orfound2:=found2;
l;


ocedure Searchx(cor1,cor2:single;var corfound:single);
e
z = array[1..2] of real;
:
1,x2,xm,z1,z2,zm,i,j1,j2,j3 : word;
found : real;
midpl,estpar,estpl,der : xz;

gin
x1:=1;x2:=po1;
z1:=1;z2:=po2;

repeat
  xm:=round((x1+x2)/2);
  zm:=round((z1+z2)/2);
  midpl[1]:=p[1]^[(xm-1)*po2+zm].pl;
  midpl[2]:=p[2]^[(xm-1)*po2+zm].pl;

  if (midpl[1]>=cor1) and (midpl[2]>=cor2) then
  begin
    x2:=xm;z2:=zm;
  end
  else if (midpl[1]>=cor1) and (midpl[2]<cor2) then
  begin
    x2:=xm;z1:=zm;
  end
  else if (midpl[1]<cor1) and (midpl[2]>=cor2) then
  begin
    x1:=xm;z2:=zm;
  end
  else if (midpl[1]<cor1) and (midpl[2]<cor2) then
  begin
    x1:=xm;z1:=zm;
  end;
until ((x2-x1)<=1) and ((z2-z1)<=1);

estpar[1]:=p[1]^[(xm-1)*po2+zm].par;
estpar[2]:=p[3]^[(xm-1)*po2+zm].par;

repeat
  estpl[1]:=0;estpl[2]:=0;der[1]:=0;der[2]:=0;
  DBasis(orderk,jlh_sta,estpar[1],x,nbasis,ndlbasis);
  DBasis(orderl,jlh_WL,estpar[2],y,mbasis,mdlbasis);
  for j1:=1 to jlh_sta do
    for j2:=1 to jlh_WL do
    begin
```

```
      estpl[1]:=estpl[1]+b[1]^[(j1-1)*jlh_WL+j2]*nbasis[1,j1]*mbasis[1,j2];
      estpl[2]:=estpl[2]+b[2]^[(j1-1)*jlh_WL+j2]*nbasis[1,j1]*mbasis[1,j2];
      der[1]:=der[1]+b[1]^[(j1-1)*jlh_WL+j2]*nd1basis[1,j1]*mbasis[1,j2];
      der[2]:=der[2]+b[2]^[(j1-1)*jlh_WL+j2]*nbasis[1,j1]*md1basis[1,j2];
    end;
   writeln(estpl[1]:9:7,'  ',estpl[2]:9:7,'  ',der[1]:9:7,'  ',der[2]:9:7);}

  estpar[1]:=estpar[1]-((estpl[1]-cor1)/der[1]);
  if estpar[1]<0 then estpar[1]:=0
    else if estpar[1]>x[jlh_sta+orderk] then estpar[1]:=x[jlh_sta+orderk];
  estpar[2]:=estpar[2]-((estpl[2]-cor2)/der[2]);
  if estpar[2]<0 then estpar[2]:=0
    else if estpar[2]>y[jlh_WL+orderl] then estpar[2]:=y[jlh_WL+orderl];

until (round(estpl[1]*1000)=round(cor1*1000)) and
round(estpl[2]*1000)=round(cor2*1000));

found:=0;
Basis(orderk,jlh_sta,estpar[1],x,nbasis);
Basis(orderl,jlh_WL,estpar[2],y,mbasis);
for j1:=1 to jlh_sta do
  for j2:=1 to jlh_WL do
    found:=found+b[3]^[(j1-1)*jlh_WL+j2]*nbasis[1,j1]*mbasis[1,j2];

corfound:=found;
d;


==========================================================}

ocedure SearchyBP(mode,nost:word; zcor:single;var found:single);
r
i,j,nw : word;
finded    : boolean;
pa,pb,pt,estpar,estpl,der : real;
temp : array[1..2]of real;

gin
if mode=1 then nw:=jlh_WL else nw:=nwlmod;
finded:=false;
i:=1;
estpar := p2d[2]^[poBP].par;
while (not finded)and(i<=poBP-1) do
begin
  pa:=p2d[2]^[i].pl;
  pb:=p2d[2]^[i+1].pl;
  if pa>pb then
  begin pt:=pa; pa:=pb; pb:=pt; end;
  if (pb>=zcor) and (pa<=zcor) then
  begin
    if p2d[2]^[i].pl=zcor then estpar:=p2d[2]^[i].par
      else if p2d[2]^[i+1].pl=zcor then estpar:=p2d[2]^[i+1].par
      else estpar := p2d[2]^[i+1].par;
    finded:=true;
  end;
  inc(i);
end;

for i:=1 to 4 do
begin
  estpl:=0;der:=0;
  DBasis(orderl,nw,estpar,y,mbasis,md1basis);
    for j:=1 to nw do
    begin
      if mode=1 then
```

```
      begin
      estpl:=estpl+b2d[2]^[(nost-1)*jlh_WL+j]*mbasis[1,j];
      der:=der+b2d[2]^[(nost-1)*jlh_WL+j]*md1basis[1,j];
      end
      else if mode=2 then
      begin
      estpl:=estpl+b2dnew[2]^[(nost-1)*nwlmod+j]*mbasis[1,j];
      der:=der+b2dnew[2]^[(nost-1)*nwlmod+j]*md1basis[1,j];
      end;
    end;

  if der<>0 then estpar:=estpar-((estpl-zcor)/der) else exit;
  if estpar<0 then estpar:=0
    else if estpar>y[nw+orderl] then estpar:=y[nw+orderl];
end;

for j:=1 to 2 do temp[j]:=0;
basis(orderl,nw,estpar,y,mbasis);
for j:=1 to nw do
  if mode=1 then
    for i:=1 to 2 do temp[i]:=temp[i]+b2d[i]^[(nost-1)*nw+j]*mbasis[1,j]
  else if mode=2 then
    for i:=1 to 2 do temp[i]:=temp[i]+b2dnew[i]^[(nost-1)*nw+j]*mbasis[1,j];

if mode=1 then found:=estpar else found:=temp[1];
id;


rocedure SearchyWL(mode,nowl:word; xcor:single;var found:single);
ar
i,j,nw : word;
finded    : boolean;
pa,pb,pt,estpar,estpl,der : real;
temp : array[1..2]of real;

egin
if mode=1 then nw:=jlh_sta else nw:=nstatmod;
finded:=false;
i:=1;estpar := p2d[1]^[poBP].par;
while (not finded) and (i<=poBP-1) do
begin
  pa:=p2d[1]^[i].pl;
  pb:=p2d[1]^[i+1].pl;
  if pa>pb then
  begin pt:=pa; pa:=pb; pb:=pt; end;

  if (pb>=xcor) and (pa<=xcor) then
  begin
    if p2d[1]^[i].pl=xcor then estpar:=p2d[1]^[i].par
      else if p2d[1]^[i+1].pl=xcor then estpar:=p2d[1]^[i+1].par
      else estpar := p2d[1]^[i+1].par;
    finded:=true;
  end;

  inc(i);
end;

for i:=1 to 4 do
begin
  estpl:=0;der:=0;
  DBasis(orderk,nw,estpar,x,nbasis,nd1basis);
    for j:=1 to nw do
    begin
      if mode=1 then
      begin
```

```
      estpl:=estpl+b2d[1]^[(j-1)*jlh_WL+nowl]*nbasis[1,j];
      der:=der+b2d[1]^[(j-1)*jlh_WL+nowl]*nd1basis[1,j];
      end
      else if mode=2 then
      begin
      estpl:=estpl+b2dnew[1]^[(j-1)*jlh_WL+nowl]*nbasis[1,j];
      der:=der+b2dnew[1]^[(j-1)*jlh_WL+nowl]*nd1basis[1,j];
      end;
    end;

  if der<>0 then estpar:=estpar-((estpl-xcor)/der) else exit;
  if estpar<0 then estpar:=0
    else if estpar>x[nw+orderk] then estpar:=x[nw+orderk];
end;

for j:=1 to 2 do temp[j]:=0;
basis(orderk,nw,estpar,x,nbasis);
for j:=1 to nw do
  if mode=1 then
    for i:=1 to 2 do temp[i]:=temp[i]+b2d[i]^[(j-1)*jlh_WL+nowl]*nbasis[1,j]
  else if mode=2 then
    for i:=1 to 2 do temp[i]:=temp[i]+b2dnew[i]^[(j-1)*jlh_WL+nowl]*nbasis[1,j];

if mode=1 then found:=estpar else found:=temp[2];
d;

{============================================================}

ocedure SearchxyBP(nost:word; zcor:single;var xfound,yfound:single);
r
i,j : word;
finded    : boolean;
pa,pb,pt,estpar,estpl,der : real;
temp : array[1..2]of real;

gin
finded:=false;
i:=1;estpar := p2d[2]^[poBP].par;
while (not finded) and (i<=poBP-1) do
begin
  pa:=p2d[2]^[i].pl;
  pb:=p2d[2]^[i+1].pl;
  if pa>pb then
  begin pt:=pa; pa:=pb; pb:=pt; end;

  if (pb>=zcor) and (pa<=zcor) then
  begin
    if p2d[2]^[i].pl=zcor then estpar:=p2d[2]^[i].par
      else if p2d[2]^[i+1].pl=zcor then estpar:=p2d[2]^[i+1].par
      else estpar := p2d[2]^[i+1].par;
    finded:=true;
  end;

  inc(i);
end;

repeat
  estpl:=0;der:=0;
  DBasis(orderl,jlh_WL,estpar,y,mbasis,md1basis);
    for j:=1 to jlh_WL do
    begin
      estpl:=estpl+b2d[3]^[(nost-1)*jlh_WL+j]*mbasis[1,j];
      der:=der+b2d[3]^[(nost-1)*jlh_WL+j]*md1basis[1,j];
    end;
```

```
  if der<>0 then estpar:=estpar-((estpl-zcor)/der) else exit;
  if estpar<0 then estpar:=0
    else if estpar>y[jlh_WL+orderl] then estpar:=y[jlh_WL+orderl];
until round(estpl*1000)=round(zcor*1000);

for j:=1 to 2 do temp[j]:=0;
basis(orderl,jlh_WL,estpar,y,mbasis);
for j:=1 to jlh_WL do
    for i:=1 to 2 do temp[i]:=temp[i]+b2d[i]^[(nost-1)*jlh_WL+j]*mbasis[1,j];

xfound:=temp[1];
yfound:=temp[2];
i;

=================================================}
procedure SearchSS(mode,nst:word;xcor:single; var found:single);
r
ar,br,i,j,xpl,xf,nost,no1 : word;
finded    : boolean;
upos,pa,pb,pt,estpar,estpl,der : real;
temp : real;
mpc  : single;
knot : knottype;

gin
finded:=false;
if mode=1 then
begin xpl:=1 ;xf:=3; end
else begin xpl:=3; xf:=1 end;

if nst=1 then
  begin upos:=0; nost:=1; no1:=1 end
else  begin upos:=x[jlh_sta+orderk]; nost:=jlh_sta; no1:=po1; end;

j:=(no1-1)*po2+1;
i:=1;estpar := p[3]^[po2].par;
while (not finded) and (i<=po2-1) do
begin
  pa:=p[xpl]^[j].pl;
  pb:=p[xpl]^[j+1].pl;
  if pa>pb then
  begin pt:=pa; pa:=pb; pb:=pt; end;

  if (pb>=xcor) and (pa<=xcor) then
  begin
    if p[xpl]^[j].pl=xcor then estpar:=p[3]^[j].par
      else if p[xpl]^[j+1].pl=xcor then estpar:=p[3]^[j+1].par
      else estpar := p[3]^[j+1].par;
    finded:=true;
  end;

  inc(i);inc(j);
end;


Basis(orderk,jlh_sta,upos,x,nbasis);
repeat
  estpl:=0;der:=0;
  DBasis(orderl,jlh_WL,estpar,y,mbasis,md1basis);
  for i:=1 to jlh_sta do
    for j:=1 to jlh_WL do
    begin
      estpl:=estpl+b[xpl]^[(nost-1)*jlh_WL+j]*mbasis[1,j]*nbasis[1,i];
      der:=der+b[xpl]^[(nost-1)*jlh_WL+j]*md1basis[1,j]*nbasis[1,i];
    end;
```

```
  if der<>0 then estpar:=estpar-((estpl-xcor)/der) else exit;
  if estpar<0 then estpar:=0
    else if estpar>y[jlh_WL+orderl] then estpar:=y[jlh_WL+orderl];
  ar:=round(estpl*1000);
  br:=round(xcor*1000);
until ar=br;

basis(orderl,jlh_WL,estpar,y,mbasis);
temp:=0;
for i:=1 to jlh_sta do
  for j:=1 to jlh_WL do
    temp:=temp+b[xf]^[(nost-1)*jlh_WL+j]*nbasis[1,i]*mbasis[1,j];

found:=temp;
i;

=====================================================}
procedure SearchSSgauss(mode,nst:word;xcor:single; var found,found2:single;var
trfound:char);
r
ar,br,i,j,xpl,xf,nost,no1 : word;
finded    : boolean;
upos,pa,pb,pt,estpar,estpl,der : real;
temp : real;
mpc  : single;
knot : knottype;

j1,j2,j3 : integer;
AA,BB,CC,QuxQw4              : single;
Qu,Qw,Quw,Quu,Qww,QuxQw          :array[1..3] of single;


gin
finded:=false;
if mode=1 then
begin xpl:=1 ;xf:=3; end
else begin xpl:=3; xf:=1 end;

if nst=1 then
  begin upos:=0; nost:=1; no1:=1 end
else  begin upos:=x[jlh_sta+orderk]; nost:=jlh_sta; no1:=po1; end;

j:=(no1-1)*po2+1;
i:=1;estpar := p[3]^[po2].par;
while (not finded) and (i<=po2-1) do
begin
  pa:=p[xpl]^[j].pl;
  pb:=p[xpl]^[j+1].pl;
  if pa>pb then
  begin pt:=pa; pa:=pb; pb:=pt; end;

  if (pb>=xcor) and (pa<=xcor) then
  begin
    if p[xpl]^[j].pl=xcor then estpar:=p[3]^[j].par
      else if p[xpl]^[j+1].pl=xcor then estpar:=p[3]^[j+1].par
      else estpar := p[3]^[j+1].par;
    finded:=true;
  end;

  inc(i);inc(j);
end;


Basis(orderk,jlh_sta,upos,x,nbasis);
```

```
epeat
  estpl:=0;der:=0;
  DBasis(orderl,jlh_WL,estpar,y,mbasis,md1basis);
  for i:=1 to jlh_sta do
    for j:=1 to jlh_WL do
    begin
      estpl:=estpl+b[xpl]^[(nost-1)*jlh_WL+j]*mbasis[1,j]*nbasis[1,i];
      der:=der+b[xpl]^[(nost-1)*jlh_WL+j]*md1basis[1,j]*nbasis[1,i];
    end;

  if der<>0 then estpar:=estpar-((estpl-xcor)/der) else exit;
  if estpar<0 then estpar:=0
    else if estpar>y[jlh_WL+orderl] then estpar:=y[jlh_WL+orderl];
  ar:=round(estpl*1000);
  br:=round(xcor*1000);
until ar=br;
temp:=0;
for j1:=1 to 3 do
begin
  qu[j1]:=0;
  qw[j1]:=0;
  quw[j1]:=0;
  quu[j1]:=0;
  qww[j1]:=0;
end;
D2Basis(orderl,jlh_WL,estpar,y,mbasis,md1basis,md2basis);
for j1:=1 to jlh_sta do
for j2:=1 to jlh_WL do
begin
  temp:=temp+b[xf]^[(nost-1)*jlh_WL+j2]*nbasis[1,j1]*mbasis[1,j2];
  for j3:=1 to 3 do
  begin
    qu[j3]:=qu[j3]+b[j3]^[(j1-1)*jlh_WL+j2]*nd1basis[1,j1]*mbasis[1,j2];
    qw[j3]:=qw[j3]+b[j3]^[(j1-1)*jlh_WL+j2]*nbasis[1,j1]*md1basis[1,j2];
    quw[j3]:=quw[j3]+b[j3]^[(j1-1)*jlh_WL+j2]*nd1basis[1,j1]*md1basis[1,j2];
    quu[j3]:=quu[j3]+b[j3]^[(j1-1)*jlh_WL+j2]*nd2basis[1,j1]*mbasis[1,j2];
    qww[j3]:=qww[j3]+b[j3]^[(j1-1)*jlh_WL+j2]*nbasis[1,j1]*md2basis[1,j2];
  end;
end;
QuxQw[1]:=qu[2]*qw[3]-qu[3]*qw[2];
QuxQw[2]:=qu[3]*qw[1]-qu[1]*qw[3];
QuxQw[3]:=qu[1]*qw[2]-qu[2]*qw[1];
QuxQw4:=sqr(sqr(QuxQw[1])+sqr(QuxQw[2])+sqr(QuxQw[3]));
AA:=QuxQw[1]*Quu[1]+QuxQw[2]*Quu[2]+QuxQw[3]*Quu[3];
BB:=QuxQw[1]*Quw[1]+QuxQw[2]*Quw[2]+QuxQw[3]*Quw[3];
CC:=QuxQw[1]*Qww[1]+QuxQw[2]*Qww[2]+QuxQw[3]*Qww[3];
if QuxQw4<>0 then
begin
  found2:=(AA*CC-sqr(BB))/QuxQw4;
  strfound:='T';
end
else
begin
  strfound:='F';found2:=0;
  if (AA*CC-sqr(BB))=0 then strfound:='0';
  if (AA*CC-sqr(BB))<0 then strfound:='-';
  if (AA*CC-sqr(BB))>0 then strfound:='+';
end;
found:=temp;
d;
==================================================}

ocedure SearchBLE(ycor:single; var found:single);
r
i,j : word;
```

```
r1,w2,wm,yest,umid : single;

gin
otaliterasi:=0;
ound:=0;
f ycor<p[2]^[(round(po1/2)-1)*po2+1].pl then exit;
1:=0; w2:=y[jlh_WL+orderl]/2;
mid:=x[jlh_sta+orderk]/2;
asis(orderk,jlh_sta,umid,x,nbasis);
epeat
  totaliterasi:=totaliterasi+1;
  wm:=(w1+w2)/2;
  yest:=0;
  Basis(orderl,jlh_WL,wm,y,mbasis);
  for i:=1 to jlh_sta do
    for j:=1 to jlh_WL do
      yest:=yest+b[2]^[(i-1)*jlh_WL+j]*nbasis[1,i]*mbasis[1,j];

  if yest>ycor then w2:=wm else w1:=wm;
ntil ((round(yest*1000)=round(ycor*1000)))or(totaliterasi>50);
found:=wm;
if totaliterasi<=50 then totaliterasi:=0;
d;

{=============================================================}

ocedure SearchU(mode:word;wpos,ycor:single;var corf1,corf2:single);
r
j1,j2 : word;
u1,u2,uest,yest : single;


gin
if mode=1 then begin u1:=0; u2:=x[jlh_sta+orderk]/2; end
  else begin u1:=x[jlh_sta+orderk]/2; u2:=x[jlh_sta+orderk]; end;

Basis(orderl,jlh_WL,wpos,y,mbasis);
repeat
  uest:=(u1+u2)/2;
  Basis(orderk,jlh_sta,uest,x,nbasis);

  yest:=0;
  for j1:=1 to jlh_sta do
    for j2:=1 to jlh_WL do
    begin
      yest:=yest+b[2]^[(j1-1)*jlh_WL+j2]*nbasis[1,j1]*mbasis[1,j2];
    end;

  case mode of
  1: if yest>ycor then u2:=uest else u1:=uest;
  2: if yest>ycor then u1:=uest else u2:=uest;
  end;
until round(yest*1000)=round(ycor*1000);

corf1:=0; corf2:=0;
Basis(orderk,jlh_sta,uest,x,nbasis);
for j1:=1 to jlh_sta do
  for j2:=1 to jlh_WL do
  begin
    corf1:=corf1+b[1]^[(j1-1)*jlh_WL+j2]*nbasis[1,j1]*mbasis[1,j2];
    corf2:=corf2+b[3]^[(j1-1)*jlh_WL+j2]*nbasis[1,j1]*mbasis[1,j2];
  end;

d;
```

```
==========================================================}

cedure Searchsk(mode:word;xcor:single;var corf:single);

1,j2 : word;
1,u2,uest,xest,wpos : single;


in
f mode=1 then wpos:=0 else wpos:=y[jlh_WL+orderl];
1:=0; u2:=x[jlh_sta+orderk];

asis(orderl,jlh_WL,wpos,y,mbasis);
epeat
  uest:=(u1+u2)/2;
  Basis(orderk,jlh_sta,uest,x,nbasis);

  xest:=0;
  for j1:=1 to jlh_sta do
    for j2:=1 to jlh_WL do
      xest:=xest+b[1]^[(j1-1)*jlh_WL+j2]*nbasis[1,j1]*mbasis[1,j2];

  if xest>xcor then u2:=uest else u1:=uest;
ntil round(xest*1000)=round(xcor*1000);

orf:=0;
asis(orderk,jlh_sta,uest,x,nbasis);
or j1:=1 to jlh_sta do
  for j2:=1 to jlh_WL do
    corf:=corf+b[3]^[(j1-1)*jlh_WL+j2]*nbasis[1,j1]*mbasis[1,j2];

;

cedure Searchblesk(mode1,mode2:word;ycor:single;var corf:single);
r
j1,j2 : word;
max,xh,u1,u2,uest,yest,wpos : single;
utside :boolean;


in
f mode1=1 then wpos:=0 else wpos:=y[jlh_WL+orderl];
f mode2=1 then begin u1:=0; u2:=x[jlh_sta+orderk]/2; end
  else begin u1:=x[jlh_sta+orderk]/2; u2:=x[jlh_sta+orderk]; end;
utside:=false;

asis(orderl,jlh_WL,wpos,y,mbasis);
f mode1=1 then
egin
  ymax:=0;
  uest:=x[jlh_sta+orderk]/2;
  Basis(orderk,jlh_sta,uest,x,nbasis);
  xh:=0;
  for j1:=1 to jlh_sta do
    for j2:=1 to jlh_WL do
    begin
      ymax:=ymax+b[2]^[(j1-1)*jlh_WL+j2]*nbasis[1,j1]*mbasis[1,j2];
      xh:=xh+b[1]^[(j1-1)*jlh_WL+j2]*nbasis[1,j1]*mbasis[1,j2];
    end;
  if ymax<ycor then
  begin
    outside:=true;
    corf:=xh;
  end;
end;
```

```
if not outside then
begin
repeat
  uest:=(u1+u2)/2;
  Basis(orderk,jlh_sta,uest,x,nbasis);

  yest:=0;
  for j1:=1 to jlh_sta do
    for j2:=1 to jlh_WL do
      yest:=yest+b[2]^[(j1-1)*jlh_WL+j2]*nbasis[1,j1]*mbasis[1,j2];

  if yest>ycor then u2:=uest else u1:=uest;
until round(yest*1000)=round(ycor*1000);

corf:=0;
Basis(orderk,jlh_sta,uest,x,nbasis);
for j1:=1 to jlh_sta do
  for j2:=1 to jlh_WL do
    corf:=corf+b[1]^[(j1-1)*jlh_WL+j2]*nbasis[1,j1]*mbasis[1,j2];
end;

d;

d.
```

```
: iodata;

erface
; ta_decl;
cedure SaveNewDataWL;
cedure SaveNewDatabd;
cedure Openline(var ok:boolean);
cedure OpenHBfB;
cedure Openpoly(var ok:boolean);
cedure SavePoly;
cedure getsignpoly;
cedure Getsignpolyx(nump:integer;var signx:integer);
cedure Getsignpolyz(nump:integer;var signz:integer);
cedure FairingPoly;

lementation

==========Write parametric data points to file==========}
cedure SaveNewDatabd;

,j,num:integer;

in
ssignfile(fl,baifile3);
I-} rewrite(fl);{$I+}
f ioresult <>0 then
egin
nd;
f ioresult <>0 then exit;
riteln(fl,'¥');
f modedata=3 then
egin
 writeln(fl,5);
 writeln(fl,jlh_sta,'  ',nwlmod);
 writeln(fl,shift);
 writeln(fl,orderk,'  ',orderl);
 for i:=1 to jlh_sta do write(fl,newstat[i],'  ');
 for i:=1 to nwlmod do write(fl,newwl[i],'  ');
 for i:=1 to jlh_sta do
 begin
 for j:=1 to nwlmod do
   begin
     num:=(i-1)*nwlmod+j;
     if dnew[2]^[num]>shipbreadth then dnew[2]^[num]:=shipbreadth;
     if j>1 then
     begin
       if dnew[2]^[num]<dnew[2]^[num-1] then dnew[2]^[num]:=dnew[2]^[num-1];
     end;
   end;
 end;
 for i:=1 to jlh_sta*nwlmod do
 begin
   for j:=1 to 3 do write(fl,dnew[j]^[i]:8:4,'  ');
   writeln(fl);
 end;
nd
lse if modedata=1 then
egin
 writeln(fl,2);
 writeln(fl,jlh_sta,'  ',nwlmod);
 writeln(fl,shift);
 writeln(fl,orderl);
 for i:=1 to nwlmod+orderl do write(fl,y[i],'  ');
 writeln(fl);
 for i:=1 to nwlmod do
```

```
 begin
   write(fl,newwl[i],'  ');
 end;
 writeln(fl);
 for i:=1 to jlh_sta*nwlmod do
 begin
   for j:=1 to 3 do write(fl,dnew[j]^[i]:8:4,'  ');
   writeln(fl);
 end;
nd;
lose(fl);
;
=======================================}
cedure SaveNewDataWL;

,j,num:integer;

in
ssignfile(fl,baifile2);
I-} rewrite(fl);{$I+}
f ioresult <>0 then
egin
nd;
f ioresult <>0 then exit;
riteln(fl,'¥');
riteln(fl,3);
riteln(fl,nstatmod,'  ',jlh_wl);
riteln(fl,shift);
riteln(fl,orderk);
or i:=1 to nstatmod+orderk do write(fl,x[i],'  ');
riteln(fl);
or i:=1 to nstatmod do
egin
  write(fl,newstat[i],'  ');
nd;
riteln(fl);
or i:=1 to nstatmod do
egin
  for j:=1 to jlh_wl do
  begin
    num:=(i-1)*jlh_wl+j;
    if dnew[2]^[num]>shipbreadth then dnew[2]^[num]:=shipbreadth;
    if j>1 then
    begin
      if dnew[2]^[num]<dnew[2]^[num-1] then dnew[2]^[num]:=dnew[2]^[num-1];
    end;
  end;
nd;
or i:=1 to nstatmod*jlh_wl do
egin
  for j:=1 to 3 do
  begin
    write(fl,(dnew[j]^[i]):8:4);
    write(fl,'    ');
  end;
  writeln(fl);
nd;
losefile(fl);
;

==========Open existed lines file=================}
ocedure Openline(var OK:boolean);
r
,i,j : integer;
code  : char;
```

```
in
ssign(fl,filelines);
$I-} reset(fl);{$I+}
f ioresult <>0 then
egin
 OK:=false;closefile(fl);exit;
nd;
eadln(fl,code);
f code<>#178 then
egin
 closefile(fl);
 OK:=false;exit;
nd;
eadln(fl,polyfilename);
eadln(fl,bdnum,bdprt);
or i:=1 to bdnum do
or j:=1 to bdprt do
egin
 for h:=1 to 3 do read(fl,bdcor[h]^[(i-1)*bdprt+j]);
 readln(fl);
nd;
eadln(fl,wlnum,wlprt);
or i:=1 to wlnum do
or j:=1 to wlprt do
egin
 for h:=1 to 3 do read(fl,wlcor[h]^[(i-1)*wlprt+j]);
 readln(fl);
nd;
eadln(fl,blnum,blprt);
or i:=1 to blnum do
or j:=1 to blprt do
egin
 for h:=1 to 3 do read(fl,blcor[h]^[(i-1)*blprt+j]);
 readln(fl);
end;
or i:=1 to 2 do
or j:=1 to bdprt do
egin
 for h:=1 to 3 do read(fl,sscor[h]^[(i-1)*bdprt+j]);
 readln(fl);
end;
closefile(fl);
OK:=true;
gaussfilename2:=filelines;
gaussfilename2[length(gaussfilename2)]:='v';
gaussfilename2[length(gaussfilename2)-1]:='c';
gaussfilename2[length(gaussfilename2)-2]:='g';
assignfile(gaussfile,gaussfilename2);
{$I-} reset(gaussfile);{$I+}
i:=0;j:=0;
for i:=1 to gnum*gprt do
egin
 read(gaussfile,gaussvar);
 if (j=0) then
 begin
   if gaussvar.warna<=0 then
   begin
     gaussminneg:=gaussvar.warna;
     gaussmaxneg:=gaussvar.warna;
     j:=1;
   end;
 end
 else if gaussvar.warna<=0 then
 begin
```

```
    if gaussvar.warna<gaussminneg then gaussminneg:=gaussvar.warna
    else if gaussvar.warna>gaussmaxneg then gaussmaxneg:=gaussvar.warna;
  end;
  if h=0 then
  begin
    if gaussvar.warna>0 then
    begin
      gaussminpos:=gaussvar.warna;
      gaussmaxpos:=gaussvar.warna;
      h:=1;
    end;
  end
  else if gaussvar.warna>0 then
  begin
    if gaussvar.warna<gaussminpos then gaussminpos:=gaussvar.warna
    else if gaussvar.warna>gaussmaxpos then gaussmaxpos:=gaussvar.warna;
  end;
  gvcor^[i].paru:=gaussvar.paru;
  gvcor^[i].parw:=gaussvar.parw;
  gvcor^[i].x:=gaussvar.x;
  gvcor^[i].y:=gaussvar.y;
  gvcor^[i].z:=gaussvar.z;
  gvcor^[i].warna:=gaussvar.warna;
nd;
entangwarnaneg:=gaussmaxneg-gaussminneg;
entangwarnapos:=gaussmaxpos-gaussminpos;
losefile(gaussfile);
l;


==========Open HBFB file================}
ocedure OpenHBfB;
:
,i,j : integer;
ode  : char;

in
ssign(fl,fileHBfB);
{$I-} reset(fl);{$I+}
f ioresult <>0 then
egin
  closefile(fl);exit;
nd;
eadln(fl,code);
f code<>#178 then
egin
  closefile(fl);exit;
nd;
eadln(fl,bdnum,bdprt);
or i:=1 to bdprt do readln(fl,lwl[i]);
or i:=1 to bdnum do
or j:=1 to bdprt do
egin
  for h:=1 to 3 do read(fl,bdcor[h]^[(i-1)*bdprt+j]);
  readln(fl);
nd;
losefile(fl);
;

==========Open existed polygon file================}
ocedure Openpoly(var OK:boolean);
:
,j:word;
code:char;
in
```

```
 K:=true;
 ssignfile(fl,polyfilename);
 $I-} reset(fl);{$I+}
 f ioresult <>0 then
 egin
  OK:=false;exit;
 nd;
 eadln(fl,code);
 f code<>#174 then
 egin
  OK:=false;
  exit;
 nd;
 eadln(fl,datafile);
 eadln(fl,jlh_sta,jlh_WL);
 eadln(fl,shift);
 eadln(fl,shipbreadth);
 eadln(fl,orderk,orderl);
 or i:=1 to jlh_sta+orderk do readln(fl,x[i]);
 or i:=1 to jlh_WL+orderl do readln(fl,y[i]);
 or i:=1 to jlh_WL*jlh_sta do
 egin
   for j:=1 to 3 do
   read(fl,b[j]^[i]);
   readln(fl);
 nd;
 losefile(fl);
 ;


 ocedure SavePoly;
 r
 ,j:word;

 gin
   assignfile(fl,polyfilename);
  {$I-} rewrite(fl);{$I+}
   if ioresult <>0 then
   begin
   end;
   if ioresult <>0 then exit;
   writeln(fl,'®');
   writeln(fl,datafile);
   writeln(fl,jlh_sta,'  ',jlh_wl);
   writeln(fl,shift);
   writeln(fl,shipbreadth);
   writeln(fl,orderk,'  ',orderl);
   for i:=1 to jlh_sta+orderk do writeln(fl,x[i]);
   for i:=1 to jlh_WL+orderl do writeln(fl,y[i]);
   for i:=1 to jlh_WL*jlh_sta do
   begin
     for j:=1 to 3 do
     write(fl,b[j]^[i]:8:4,'  ');
     writeln(fl);
   end;
   closefile(fl);
 d;


 ocedure Getsignpolyx(nump:integer;var signx:integer);
 r
 i2,j2,nump1,nump2,nump3:integer;
 tfound,yfound:single;
 gin
 i2:=((nump-1)div jlh_WL)+1;
```

```
2:=((nump-1)mod jlh_WL)+1;
f i2=1 then i2:=2;
f i2=jlh_sta then i2:=jlh_sta-1;
ump2:=(i2-1)*jlh_WL+j2;
ump1:=(i2-2)*jlh_WL+j2;
ump3:=(i2)*jlh_WL+j2;
found:=(b[1]^[nump2]-b[1]^[nump1])/(b[1]^[nump3]-b[1]^[nump1]);
found:=b[2]^[nump1]+(b[2]^[nump3]-b[2]^[nump1])*tfound;
f yfound<b[2]^[nump2] then signx:=1
lse begin
 if yfound>b[2]^[nump2] then signx:=-1
 else signx:=0;
nd;
;

cedure Getsignpolyz(nump:integer;var signz:integer);

2,j2,nump1,nump2,nump3:integer;
found,yfound:single;
in
2:=((nump-1)div jlh_WL)+1;
2:=((nump-1)mod jlh_WL)+1;
f j2=1 then j2:=2;
f j2=jlh_WL then j2:=jlh_WL-1;
ump2:=(i2-1)*jlh_WL+j2;
ump1:=(i2-1)*jlh_WL+j2-1;
ump3:=(i2-1)*jlh_WL+j2+1;
found:=(b[3]^[nump2]-b[3]^[nump1])/(b[3]^[nump3]-b[3]^[nump1]);
found:=b[2]^[nump1]+(b[2]^[nump3]-b[2]^[nump1])*tfound;
f yfound<b[2]^[nump2] then signz:=1
lse begin
 if yfound>b[2]^[nump2] then signz:=-1
 else signz:=0;
nd;
;

cedure getsignpoly;

i,signx,signz:integer;
in
or i:=1 to jlh_sta*jlh_WL do
egin
 getsignpolyx(i,signx);
 getsignpolyz(i,signz);
 bsignx[i]:=signx;
 bsignz[i]:=signz;
nd;
;


cedure FairingPoly;
 i:integer;
in
or i:=1 to jlh_sta*jlh_wl do
egin
 if b[2]^[i]>shipbreadth then b[2]^[i]:=shipbreadth;
nd;
;


.
```

```
t bsplfit;

erface
s ta_decl,basknot,bspl2d,view2d;
cedure Bspline;
cedure Opendat(s:string;var mat :pointtyp3);
cedure BodyPlan;
cedure WLPlan;
cedure Bfit;
cedure Paramu;
cedure Paramw;

lementation
s calcpol3;
==============open data file===============}
cedure Opendat(s:string;var mat :pointtyp3);
:
,j,num,ext1,ext2,nnwl,numdf : word;
if      : string;
emp     : real;
ode:char;
in
assignfile(fl,s);
{$I-} reset(fl);{$I+}
if ioresult <>0 then
begin
end;
if ioresult <>0 then exit;
readln(fl,code);
new(stat);new(wl);
readln(fl,modedata);
if modedata=1 then
begin
  new(maxb);
  readln(fl,jlh_sta,jlh_WL);
  readln(fl,wsh,wk);
  if wsh=1 then nnwl:=jlh_WL-1 else nnwl:=jlh_WL;
  for i:=0 to nnwl do read(fl,maxb^[i]);
  for i:=0 to nnwl do read(fl,wl^[i]);
  for i:=1 to jlh_sta do
  begin
     read(fl,stat^[i]);
     for j:=1 to nnwl do
     begin
       num:=(i-1)*jlh_WL+j;
       read(fl,mat[2,num]);
       mat[2,num]:=mat[2,num]*maxb^[j];
       mat[1,num]:=stat^[i];
       mat[3,num]:=wl^[j];
     end;
  end;
  for i:=1 to jlh_WL do
    readln(fl,ssx[1,i],ssx[2,i]);

  if wsh=1 then
  begin
    new(sheer[1]);new(sheer[2]);
    readln(fl,ssz[1],ssz[2]);
    for i:=1 to jlh_sta do
      read(fl,sheer[1]^[i]);
    wl^[jlh_WL]:=sheer[1]^[jlh_wl];
    readln(fl);
    for i:=1 to jlh_sta do
      read(fl,sheer[2]^[i]);
    readln(fl);
```

```
    for i:=1 to jlh_sta do
    begin
      mat[1,i*jlh_WL]:=stat^[i];
      mat[2,i*jlh_WL]:=sheer[2]^[i];
      mat[3,i*jlh_WL]:=sheer[1]^[i];
    end;
    for i:=1 to 2 do dispose(sheer[i]);
  end;
  if wk=1 then
  begin
    new(keel);
    for i:=1 to jlh_sta do
    begin
      read(fl,keel^[i]);
      mat[3,(i-1)*jlh_WL+1]:=mat[3,(i-1)*jlh_WL+1]+keel^[i];
    end;
    dispose(keel);
  end;

  dispose(maxb);
  closefile(fl);

rt data}
  if stat^[1]>stat^[2] then
  begin
    for i:=1 to trunc(jlh_sta/2) do
    begin
      temp:=stat^[i];
      stat^[i]:=stat^[jlh_sta-i+1];
      stat^[jlh_sta-i+1]:=temp;
    end;
    for i:=1 to jlh_WL do
      for j:=1 to trunc(jlh_sta/2) do
      begin
        temp:=mat[2,(j-1)*jlh_WL+i];
        mat[2,(j-1)*jlh_WL+i]:=mat[2,(jlh_sta-j)*jlh_WL+i];
        mat[2,(jlh_sta-j)*jlh_WL+i]:=temp;
        temp:=mat[1,(j-1)*jlh_WL+i];
        mat[1,(j-1)*jlh_WL+i]:=mat[1,(jlh_sta-j)*jlh_WL+i];
        mat[1,(jlh_sta-j)*jlh_WL+i]:=temp;
      end;
  end;

  if wl^[1]>wl^[2] then
  begin
    for i:=1 to trunc(jlh_WL/2) do
    begin
      temp:=wl^[i];
      wl^[i]:=wl^[jlh_WL-i+1];
      wl^[jlh_WL-i+1]:=temp;
    end;
    for i:=1 to jlh_sta do
      for j:=1 to trunc(jlh_WL/2) do
      begin
        temp:=mat[2,(i-1)*jlh_WL+j];
        mat[2,(i-1)*jlh_WL+j]:=mat[2,(i-1)*jlh_WL+(jlh_WL-j+1)];
        mat[2,(i-1)*jlh_WL+(jlh_WL-j+1)]:=temp;
        temp:=mat[3,(i-1)*jlh_WL+j];
        mat[3,(i-1)*jlh_WL+j]:=mat[3,(i-1)*jlh_WL+(jlh_WL-j+1)];
        mat[3,(i-1)*jlh_WL+(jlh_WL-j+1)]:=temp;
      end;
  end;
nd

lse
```

```
  begin
    readln(fl,jlh_sta,jlh_WL);
    readln(fl,shift);
    if modedata=2 then
    begin
      readln(fl,orderl);
      for i:=1 to jlh_WL+orderl do read(fl,y[i]);
      for i:=1 to jlh_WL do read(fl,newwl[i]);
    end
    else if modedata=3 then
    begin
      readln(fl,orderk);
      for i:=1 to jlh_sta+orderk do read(fl,x[i]);
      for i:=1 to jlh_sta do read(fl,newstat[i]);
    end

    else if modedata=5 then
    begin
      readln(fl,orderk,orderl);
      for i:=1 to jlh_sta do read(fl,newstat[i]);
      for i:=1 to jlh_WL do read(fl,newwl[i]);
    end;

    for i:=1 to jlh_sta*jlh_WL do
      begin
      for j:=1 to 3 do
        read(fl,mat[j,i]);
      end;
    closefile(fl);
  end;
end;


{======================Body Plan 2 Dimension=================}
Procedure BodyPlan;
var
  i: word;
  xsc,ysc,heig,bread: single;

begin
  opendat(baifile2,tempd);
  new(wpar);
  Paramw;
  for i:=1 to jlh_wl do hmg[i]:=1;
  Bfit2d(mode);
  heig:=tempd[3,jlh_sta*jlh_wl];
  bread:=2*tempd[2,round(jlh_sta/2)*jlh_wl];
  xsc:=540/bread;ysc:=330/heig;
  if xsc>ysc then scale:=ysc else scale:=xsc;
  y0:=300;z0:=412;
  poBP:=50;showerr:=false;
  igen:=1;
  if (modedata=3) then
  begin
    if mode=1 then
    begin
      igen:=2;
      for i:=1 to 3 do new(b2dnew[i]);
      Bfit2dmod;
    end;
  end;
end;
```

```
{======================Water Line Plan 2 Dimension=================}
Procedure WLPlan;
var
  i : word;
  loa : single;

begin
  opendat(baifile,tempd);
  for i:=1 to jlh_wl do hmg[i]:=1;
  shift:=0;
  Bfit2dwl(mode);
  loa:=tempd[1,jlh_sta*jlh_wl]-tempd[1,jlh_wl]-ssx[1,jlh_wl]+ssx[2,jlh_wl];
  scale:=600/loa;
  y0:=300;x0:=70;
  igen:=1;
  poBP:=80;
  if modedata=1 then
  begin
    if (mode=1)then
    begin
      Bfit2dmodwl;
      loa:=dnew[1]^[nstatmod*jlh_wl]-dnew[1]^[jlh_wl];
      scale:=620/loa;
      showerr:=true;
      igen:=2;
    end;
  end;
end;

{============main program of Bspline===========}
Procedure Bspline;
var
   g,h,i,j,jl,s,icon : word;
   u,w,u1,u2,w1,w2,wplus,uplus : single;
begin
  u1:=0;u2:=1;w1:=0;w2:=1;
  nplusc := jlh_sta + orderk;
  mplusc := jlh_WL + orderl;
  for i:=1 to jlh_WL do hmg[i]:=1;
  for i:=1 to 3 do
  begin
    new(p[i]);
    for j:=1 to po1*po2 do
    begin
      p[i]^[j].pl:=0;
      p[i]^[j].par:=0;
    end;
  end;
  icon:=0;
  u1:=u1*(jlh_sta-orderk+1);u2:=u2*(jlh_sta-orderk+1);
  w1:=w1*(jlh_WL-orderl+1);w2:=w2*(jlh_WL-orderl+1);

  u:=u1;
  uplus:=(u2-u1)/(po1-1);
  wplus:=(w2-w1)/(po2-1);
  for g:=1 to po1 do
  begin
    D2Basis(orderk,jlh_sta,u,x,nbasis,ND1BASIS,ND2BASIS);
    w:=w1;
    for h:=1 to po2 do
    begin
      D2Basis(orderl,jlh_WL,w,y,mbasis,MD1BASIS,MD2BASIS);
      inc(icon,1);
      p[1]^[icon].par:=u;
      p[2]^[icon].par:=0;
```

```
      p[3]^[icon].par:=w;
      for i:=1 to jlh_sta do
      for j:=1 to jlh_WL do
      begin
        jl:=jlh_WL*(i-1)+j;
        for s:=1 to 3 do
        begin
          p[s]^[icon].pl:=p[s]^[icon].pl+b[s]^[jl]*nbasis[1,i]*mbasis[1,j];
        end;
      end;
      w:=w1+(wplus*h);
    end;
    u:=u1+(uplus*g);
  end;
  showdat:=false;
end;

{======Calculate parameter value based on chord length======}
Procedure Paramw;
var
  i,j,dnum : word;
  isum,sum : single;

begin
  for i:=1 to jlh_sta*jlh_WL do wpar^[i]:=0;
  for j:=1 to jlh_sta do
  begin
    dnum:=(j-1)*jlh_WL;
    sum:=0;isum:=0;
    wpar^[dnum+1]:= 0;

    for i:=dnum+2 to dnum+jlh_WL do
    sum:=sum+ sqrt(sqr(tempd[1,i]-tempd[1,i-1])+sqr(tempd[2,i]-tempd[2,i-1])+
         sqr(tempd[3,i]-tempd[3,i-1]));
    for i:=dnum+2 to dnum+jlh_WL do
    begin
      isum:=isum+sqrt(sqr(tempd[1,i]-tempd[1,i-1])+sqr(tempd[2,i]-tempd[2,i-1])+
            sqr(tempd[3,i]-tempd[3,i-1]));
      wpar^[i]:=isum/sum;
    end;
  end;{j}
end;

Procedure Paramu;
var
  i,j,dnum : word;
  isum,sum : single;

begin
  for i:=1 to jlh_sta*jlh_WL do upar^[i]:=0;
  for j:=1 to jlh_WL do
  begin
    sum:=0;isum:=0;
    upar^[j]:= 0;
    for i:=2 to jlh_sta do
    begin
      dnum:=(i-1)*jlh_WL+j;
      sum:=sum+sqrt(sqr(tempd[1,dnum]-tempd[1,dnum-jlh_WL])+
           sqr(tempd[2,dnum]-tempd[2,dnum-jlh_WL])+
           sqr(tempd[3,dnum]-tempd[3,dnum-jlh_WL]));
    end;
    for i:=2 to jlh_sta do
    begin
      dnum:=(i-1)*jlh_WL+j;
      isum := isum+sqrt(sqr(tempd[1,dnum]-tempd[1,dnum-jlh_WL]) +
```

```
                sqr(tempd[2,dnum]-tempd[2,dnum-jlh_WL])+
                sqr(tempd[3,dnum]-tempd[3,dnum-jlh_WL]));
        upar^[dnum]:=isum/sum;
      end;
    end;
end;



{================Bspline curve fitting============}
Procedure Bfit;
var
  h,i,j : word;
  nfit : array[1..maxpts] of ^fit3type;
   callpaint:boolean;
   totalhit:integer;
   hit:integer;

Procedure Gauss(ar,ac,br,bc : word);
{gauss from bai tanpa kurungan}
var
   r,s,i,act : word;
   kons1,kons2 : single;
begin
  callpaint:=false;
  noprogress:=2;poly3.repaint;
  noprogress:=2;poly3.repaint;
  act:=ac+bc;hit:=0;
  for r:= 1 to ar do
  for s:=1 to bc do nfit[r]^[ac+s] := tempd[s,r];
  for r:= 1 to ar do
  begin
    kons1:=nfit[r]^[r];
    for s := 1 to act do nfit[r]^[s] := nfit[r]^[s]/kons1;
    for s := 1 to ar do
    begin
      if s<>r then
      begin
        kons2:= nfit[s]^[r];
        for i:= 1 to act do
        nfit[s]^[i] := nfit[s]^[i]-(kons2*nfit[r]^[i]);
      end;
    end;
    if hit<round((r)*100/totalhit) then callpaint:=true;
    hit:=round((r)*100/totalhit);
    if callpaint then
    begin
      poly3.Gauge1.AddProgress(1);
      callpaint:=false;
    end;
  end;
  for r:= 1 to ar do
  for s:=1 to bc do b[s]^[r]:=nfit[r]^[ac+s];
end;


begin
  nplusc := jlh_sta + orderk;
  mplusc:= jlh_wl+orderl;
  for i:=1 to jlh_wl do hmg[i]:=1;
  case modedata of
  1:
  begin
    openknot(jlh_sta,orderk,x);
    knotp(jlh_wl,orderl,y);
```

```
    end;
    2:openknot(jlh_sta,orderk,x);
    3:knotp(jlh_wl,orderl,y);
    5:
    begin
      knotpar(jlh_sta,orderk,newstat,x);
      knotpar(jlh_wl,orderl,newwl,y);
    end;
    else
    end;
    new(upar);new(wpar);
    Paramu;
    Paramw;
    callpaint:=false;hit:=0;
    for i:=1 to jlh_sta*jlh_wl do new(nfit[i]);
    totalhit:=jlh_sta*jlh_wl;
    noprogress:=1;poly3.repaint;
    noprogress:=1;poly3.repaint;
    for i:=1 to jlh_sta*jlh_wl do
    begin
      if hit<round(i*100/totalhit) then callpaint:=true;
      hit:=round(i*100/totalhit);
      if callpaint then
      begin
        poly3.Gauge1.AddProgress(1);
        callpaint:=false;
      end;
      u:= upar[i]*x[nplusc];
      w:= wpar[i]*y[mplusc];
      basis(orderk,jlh_sta,u,x,nbasis);
      rbasis(orderl,jlh_wl,w,y,hmg,mbasis);
      for j:=1 to jlh_sta do
      for h:=1 to jlh_wl do
      nfit[i]^[(j-1)*jlh_wl+h]:=nbasis[1,j]*mbasis[1,h];
      if nfit[i]^[i]=0 then
      begin
      end;
    end;
     dispose(upar);dispose(wpar);
     gauss(jlh_sta*jlh_wl,jlh_sta*jlh_wl,jlh_sta*jlh_wl,3);
     for i:=1 to jlh_wl*jlh_sta do dispose(nfit[i]);
end;



end.
```

```
unit basknot;

interface
uses ta_decl;
Procedure OpenKnot(pts,kl:integer; var mknot:knottype);
Procedure knotc(pts,kl:integer; polytemp:polytt; var cknot:knottype);
Procedure knotpar(pts,kl:integer; chordpar:stattype; var cknot:knottype);
Procedure knotc2(pts,kl:integer; var cknot:knottype);
Procedure knotp(pts,kl:integer; var pknot:knottype);
Procedure Basis(kl,pts:integer ; t:single; mknot:knottype; var bas:mattype);
Procedure Rbasis(kl,pts:integer ; t:single; mknot:knottype; h:stattype; var
 bas:mattype);
Procedure DBasis(kl,pts:integer ; t:single; mknot:knottype; var bas,dbas:mattype);
Procedure D2basis(kl,npts:integer;t:single;mknot:knottype;var bas,dbas1,dbas2:mattype);

implementation

type
  co3dtype = array[1..3]of single;

var
  uarea : array[1..maxstat]of single;
  warea : array[1..maxwl]of single;

{========calculate open knot vector============}
Procedure OpenKnot(pts,kl:integer; var mknot:knottype);
var
  i:integer;
begin
   mknot[1]:=0;
   for i:=2 to pts+kl do
      if (i>kl) and (i<(pts+2)) then mknot[i]:= mknot[i-1] + 1
         else mknot[i] := mknot[i-1];
end;

{=========calculate non uniform open knot vector==================}
{=========proportional to the chord lengths======================}

Procedure knotc(pts,kl:integer; polytemp:polytt; var cknot:knottype);
var
  i,j,n :integer;
  maxchord,sum : single;
  chord : array[1..maxstat] of single;

begin
   n:=pts-1;
  maxchord:=0;
  for i:=2 to pts do
  begin

 chord[i-1]:=sqrt(sqr(polytemp[1]^[i]-polytemp[1]^[i-1])+sqr(polytemp[2]^[i]-polytemp[2
]^[i-1])+
      sqr(polytemp[3]^[i]-polytemp[3]^[i-1]));
    maxchord:=maxchord+chord[i-1];
  end;

  for i:=1 to kl do
    cknot[i]:=0;

  for i:=1 to n-kl+1 do
  begin
    sum:=0;
    for j:=1 to i do
      sum:=sum+chord[j];
    cknot[kl+i]:=((i/(n-kl+2))*chord[i+1] + sum)/maxchord*(n-kl+2);
```

```
  end;

  for i:=n+2 to pts+kl do
    cknot[i]:=n-kl+2;
end;

{=========================================================}

Procedure knotpar(pts,kl:integer; chordpar:stattype; var cknot:knottype);
var
  i,j,n :integer;
  maxchord,sum : single;
  chord : array[1..maxstat] of single;

begin
  n:=pts-1;
  maxchord:=1;
  for i:=2 to pts do
    chord[i-1]:=(chordpar[i]-chordpar[i-1]);

  for i:=1 to kl do
    cknot[i]:=0;

  for i:=1 to n-kl+1 do
  begin
    sum:=0;
    for j:=1 to i do
      sum:=sum+chord[j];
    cknot[kl+i]:=((i/(n-kl+2))*chord[i+1] + sum)/maxchord*(n-kl+2);
  end;

  for i:=n+2 to pts+kl do
    cknot[i]:=n-kl+2;
end;

Procedure knotc2(pts,kl:integer; var cknot:knottype);
var
  bul,i,n :integer;
  pplus,pnow,deci,maxchord : single;

begin
  n:=pts-1;
  maxchord:=wl^[jlh_WL];

  for i:=1 to kl do
    cknot[i]:=0;

  pplus:=n/(n-kl+2);
  for i:=1 to n-kl+1 do
  begin
    pnow:=pplus*i;
    bul:=trunc(pnow);
    deci:=frac(pnow);
    cknot[kl+i]:=(deci*(wl^[bul+2]-wl^[bul+1]) + wl^[bul+1])/maxchord*(n-kl+2);
  end;

  for i:=n+2 to pts+kl do
    cknot[i]:=n-kl+2;
end;

{===========Calculate Triangle area in space=====================}
Function TriArea(c1,c2,c3 : co3dtype): single;
var
  al,bl,cl,s : single;
```

```
begin
  al:=sqrt(sqr(c1[1]-c2[1])+sqr(c1[2]-c2[2])+sqr(c1[3]-c2[3]));
  bl:=sqrt(sqr(c2[1]-c3[1])+sqr(c2[2]-c3[2])+sqr(c2[3]-c3[3]));
  cl:=sqrt(sqr(c3[1]-c1[1])+sqr(c3[2]-c1[2])+sqr(c3[3]-c1[3]));
  s:= (al+bl+cl)/2;
  TriArea:=sqrt(s*(s-al)*(s-bl)*(s-cl));
end;

{===========Calculate Surface Area==============================}
Function SurfArea : single;
type
  areatype= array[1..12000]of single;
var
  h,i,j,d1,d2,d3 : integer;
  area  : ^areatype;
  temparea,tempsurf : single;
  c1,c2,c3 : co3dtype;

begin
  new(area);
  tempsurf:=0;
  for i:=1 to jlh_sta-1 do
    for j:=1 to jlh_WL-1 do
    begin
      d1:=(i-1)*jlh_WL+j; d2:=d1+1; d3:=d2+jlh_WL;
      for h:=1 to 3 do c1[h]:=tempd[h,d1];
      for h:=1 to 3 do c2[h]:=tempd[h,d2];
      for h:=1 to 3 do c3[h]:=tempd[h,d3];
      temparea:=TriArea(c1,c2,c3);

      for h:=1 to 3 do c3[h]:=tempd[h,d3];
      temparea:=temparea + TriArea(c1,c2,c3);

      tempsurf:=tempsurf+temparea;
      area^[(i-1)*(jlh_WL-1)+j]:=temparea;
    end;

  SurfArea:=Tempsurf;

  for i:=1 to jlh_sta-1 do
  begin
    temparea:=0;
    for j:=1 to jlh_WL-1 do
      temparea:=temparea+area^[(i-1)*(jlh_WL-1)+j];
    uarea[i]:=temparea;
  end;

  for i:=1 to jlh_WL-1 do
  begin
    temparea:=0;
    for j:=1 to jlh_sta-1 do
      temparea:=temparea+area^[(j-1)*(jlh_WL-1)+i];
    warea[i]:=temparea;
  end;

  dispose(area);
end;

{=========calculate non uniform open knot vector=================}
{=========proportional to the chord lengths======================}

Procedure knotp(pts,kl:integer; var pknot:knottype);
var
  bul,i,j,n :integer;
  pplus,pnow,deci,maxarea,sum,numer : single;
```

```
begin
  n:=pts-1;
  maxarea:=SurfArea;
  numer:=0;
  for i:=1 to kl do
    pknot[i]:=0;

  pplus:=n/(n-kl+2);
  for i:=1 to n-kl+1 do
  begin
    pnow:=i*pplus;
    bul:=trunc(pnow);
    deci:=frac(pnow);
    sum:=0;
    if pts=jlh_WL then
    begin
      for j:=1 to i do
        sum:=sum+warea[j];
      numer:=deci*warea[bul+1] + sum;
    end
    else if pts=jlh_sta then
    begin
      for j:=1 to i do
        sum:=sum+uarea[j];
      numer:=deci*uarea[bul+1] + sum;
    end;

    pknot[kl+i]:=(numer/maxarea)*(n-kl+2);
  end;

  for i:=n+2 to pts+kl do
    pknot[i]:=n-kl+2;
end;

{======B-Spline basis function for open uniform knot vector===}
Procedure Basis(kl,pts:integer ; t:single; mknot:knottype; var bas:mattype);
var
   d,e : single;
   temp2 : array[1..maxstat] of single;
   pplusk,ic,c : integer;

begin
   for ic:=1 to pts+kl do
   begin
     temp2[ic]:=0;
   end;
   pplusk:=pts+kl;
   for ic:= 1 to pplusk-1 do
      if (t >= mknot[ic]) and (t < mknot[ic+1]) then temp2[ic] := 1
         else temp2[ic] := 0;
   for c:= 2 to kl do
      for ic := 1 to pplusk-c do
      begin
         if temp2[ic]<>0 then d := ((t-mknot[ic])*temp2[ic])/(mknot[ic+c-1]-mknot[ic])
            else d:= 0;
         if temp2[ic+1]<>0 then e :=
 ((mknot[ic+c]-t)*temp2[ic+1])/(mknot[ic+c]-mknot[ic+1])
            else e:=0;
         temp2[ic]:= d+e;
      end;
   if t >= mknot[pplusk] then temp2[pts] := 1;
   for ic := 1 to pts do bas[1,ic]:=temp2[ic];
   if t >= mknot[pplusk] then bas[1,pts]:=1;
end;
```

```
{======Rational B-Spline basis function for open knot vector==========}

Procedure Rbasis(kl,pts:integer ; t:single; mknot:knottype; h:stattype; var
 bas:mattype);
var
   sum,d,e : single;
   temp2 : array[1..maxstat] of single;
   pplusk,ic,c : integer;

begin
   for ic:=1 to pts+kl do
   begin
     temp2[ic]:=0;
   end;
   pplusk:=pts+kl;
   for ic:= 1 to pplusk-1 do
      if (t >= mknot[ic]) and (t < mknot[ic+1]) then temp2[ic] := 1
         else temp2[ic] := 0;
   for c:= 2 to kl do
      for ic := 1 to pplusk-c do
      begin
         if temp2[ic]<>0 then d := ((t-mknot[ic])*temp2[ic])/(mknot[ic+c-1]-mknot[ic])
            else d:= 0;
         if temp2[ic+1]<>0 then e :=
 ((mknot[ic+c]-t)*temp2[ic+1])/(mknot[ic+c]-mknot[ic+1])
            else e:=0;
         temp2[ic]:= d+e;
      end;
   if t >= mknot[pplusk] then temp2[pts] := 1;

   sum:=0;
   for ic:=1 to pts do
     sum:=sum+(temp2[ic]*h[ic]);

   for ic:=1 to pts do
     if sum<>0 then
       bas[1,ic]:=temp2[ic]*h[ic]/sum
     else
       bas[1,ic]:=0;

   if t >= mknot[pplusk] then bas[1,pts]:=1*h[pts]/sum;
end;


{========B-Spline basis function and the first derivative==========}

Procedure DBasis(kl,pts:integer ; t:single; mknot:knottype; var bas,dbas:mattype);
var
   b1,b2,d1,d2,d3,d4 : single;
   temp1,temp2 : array[1..maxstat] of single;
   pplusk,ic,c : integer;

begin
   for ic:=1 to pts+kl do
   begin
     temp1[ic]:=0;
     temp2[ic]:=0;
   end;
   pplusk:=pts+kl;
   for ic:= 1 to pplusk-1 do
      if (t >= mknot[ic]) and (t < mknot[ic+1]) then temp1[ic] := 1
         else temp1[ic] := 0;
   if t>=mknot[pplusk] then temp1[pts]:=1;
   for c:= 2 to kl do
```

```
      for ic := 1 to pplusk-c do
      begin
         if temp1[ic]<>0 then b1:= ((t-mknot[ic])*temp1[ic])/(mknot[ic+c-1]-mknot[ic])
            else b1:= 0;
         if temp1[ic+1]<>0 then b2:=
 ((mknot[ic+c]-t)*temp1[ic+1])/(mknot[ic+c]-mknot[ic+1])
            else b2:=0;

         if temp1[ic]<>0 then d1:=temp1[ic]/(mknot[ic+c-1]-mknot[ic])
           else d1:=0;
         if temp1[ic+1]<>0 then d2:=-temp1[ic+1]/(mknot[ic+c]-mknot[ic+1])
           else d2:=0;
         if temp2[ic]<>0 then d3:=((t-mknot[ic])*temp2[ic])/(mknot[ic+c-1]-mknot[ic])
           else d3:=0;
         if temp2[ic+1]<>0 then d4:=
 ((mknot[ic+c]-t)*temp2[ic+1])/(mknot[ic+c]-mknot[ic+1])
            else d4:=0;

         temp1[ic]:= b1+b2;
         temp2[ic]:= d1+d2+d3+d4;
      end;

   if t >= mknot[pplusk] then temp1[pts] := 1;
   for ic := 1 to pts do bas[1,ic]:=temp1[ic];
{   if t >= mknot[pplusk] then bas[1,pts]:=1;}

   for ic := 1 to pts do dbas[1,ic]:=temp2[ic];
end;

Procedure D2basis(kl,npts:integer;t:single;mknot:knottype;var bas,dbas1,dbas2:mattype);
var
   b1,b2,d1,d2,d3,d4,s1,s2,s3,s4 : single;
   temp,temp1,temp2 : array[1..100] of single;
   pplusk,ic,c : integer;

begin
   pplusk:=npts+kl;
   for ic:=1 to npts+kl do
   begin
     temp[ic]:=0;
     temp1[ic]:=0;
     temp2[ic]:=0;
   end;
   {calculate first order basis function}
   for ic:= 1 to pplusk-1 do
      if (t >= mknot[ic]) and (t < mknot[ic+1]) then temp[ic] := 1
         else temp[ic] := 0;
   if t>=mknot[pplusk] then temp[npts]:=1;
   {calculate higher order basis function and their derivatives}
   for c:= 2 to kl do
   begin
     for ic := 1 to pplusk-c do
     begin
       {calculate basis function}
       if temp[ic]<>0 then
          b1:= ((t-mknot[ic])*temp[ic])/(mknot[ic+c-1]-mknot[ic])
       else b1:= 0;
       if temp[ic+1]<>0 then
          b2:=((mknot[ic+c]-t)*temp[ic+1])/(mknot[ic+c]-mknot[ic+1])
       else b2:=0;
       {calculate first derivative}
       if temp[ic]<>0 then
         d1:=temp[ic]/(mknot[ic+c-1]-mknot[ic])
       else d1:=0;
       if temp[ic+1]<>0 then
```

```
        d2:=-temp[ic+1]/(mknot[ic+c]-mknot[ic+1])
      else d2:=0;
      if temp1[ic]<>0 then
        d3:=((t-mknot[ic])*temp1[ic])/(mknot[ic+c-1]-mknot[ic])
      else d3:=0;
      if temp1[ic+1]<>0 then
         d4:= ((mknot[ic+c]-t)*temp1[ic+1])/(mknot[ic+c]-mknot[ic+1])
      else d4:=0;
      {calculate SECOND derivative}
      if temp1[ic]<>0 then
        s1:=2*temp1[ic]/(mknot[ic+c-1]-mknot[ic])
      else s1:=0;
      if temp1[ic+1]<>0 then
        s2:=-2*temp1[ic+1]/(mknot[ic+c]-mknot[ic+1])
      else s2:=0;
      if temp2[ic]<>0 then
        s3:=(t-mknot[ic])*temp2[ic]/(mknot[ic+c-1]-mknot[ic])
      else s3:=0;
      if temp2[ic+1]<>0 then
         s4:= (mknot[ic+c]-t)*temp2[ic+1]/(mknot[ic+c]-mknot[ic+1])
      else s4:=0;

      temp[ic]:=b1+b2;
      temp1[ic]:=d1+d2+d3+d4;
      temp2[ic]:=s1+s2+s3+s4;
    end;
  end;
  {put in arrays}
  for ic:=1 to npts do
  begin
    bas[1,ic]:=temp[ic];
    dbas1[1,ic]:=temp1[ic];
    dbas2[1,ic]:=temp2[ic]
  end;
end;

end.
```

```
unit bspl2d;

interface
uses basknot,ta_decl;
Procedure Bfit2dwl(mode:word);
Procedure BsplWl(iwl,num,po,mode:word; xknot:knottype);
Procedure Bfit2dmodwl;
Procedure Bsplbody(istat,num,po,mode:word);
Procedure Bfit2dmod;
Procedure Bfit2d(mode:word);
Procedure GetNewDat(istat:word);

implementation
uses bsplfit;

type
  partype = array[1..50] of single;
  fit2dtype = array[1..maxstat] of single;
var
  dtemp  : array[1..3,1..maxstat] of single;
  par    : ^partype;
  nfit2d : array[1..maxstat] of ^fit2dtype;


{=========================Gauss Elimination 2D====================}
Procedure Gauss2d(ar,ac,br,bc : word);
var
   r,s,i,act : word;
   kons1,kons2 : single;

begin
  act:=ac+bc;
  for r:= 1 to ar do
  begin
   if nfit2d[r]^[r]=0 then
   begin
     i:=r+1;
     while (nfit2d[i]^[i]=0)and(i<=ar) do
     begin
       i:=i+1;
     end;
     for s:=1 to act do
     begin
       nfit2d[r]^[s]:=nfit2d[r]^[s]+nfit2d[i]^[s];
     end;
   end;
   kons1:=nfit2d[r]^[r];
   if kons1=0 then kons1:=1e-7;
   for s := 1 to act do
      nfit2d[r]^[s] := nfit2d[r]^[s]/kons1;
   for s := 1 to ar do
   if s<>r then
   begin
     kons2:= nfit2d[s]^[r];
     for i:= 1 to act do
     nfit2d[s]^[i] := nfit2d[s]^[i] - (kons2 * nfit2d[r]^[i]);
   end;
 end;
end;

{=============================================================================}

Procedure BsplWl(iwl,num,po,mode:word; xknot:knottype);
var
   g,i,s : word;
```

```
   uplus,u : real;

begin
   u:=0;
   uplus:=xknot[num+orderk]/(po-1);
   for g:=1 to po do
   begin
     basis(orderk,num,u,xknot,nbasis);
     p2d[1]^[g].par:=0;
     p2d[2]^[g].par:=u;
     for s:=1 to 2 do p2d[s]^[g].pl:=0;
     for i:=1 to num do
       for s:=1 to 2 do
         if mode=1 then
 p2d[s]^[g].pl:=p2d[s]^[g].pl+b2d[s]^[(i-1)*jlh_wl+iwl]*nbasis[1,i]
          else p2d[s]^[g].pl:=p2d[s]^[g].pl+b2dnew[s]^[(i-1)*jlh_wl+iwl]*nbasis[1,i];
     u:= uplus*g;
   end;
end;



{============================================================}

Procedure Paruperwl(num:word);
var
  i : word;
  isum,sum : real;

begin
  for i:=1 to num do par^[i]:=0;
    sum:=0;isum:=0;
    par^[1]:= 0;

    for i:=2 to num do
      sum:=sum+ sqrt(sqr(dtemp[1,i]-dtemp[1,i-1]) + sqr(dtemp[2,i]-dtemp[2,i-1])
           + sqr(dtemp[3,i]-dtemp[3,i-1]));
    for i:=2 to num do
    begin
      isum := isum+ sqrt(sqr(dtemp[1,i]-dtemp[1,i-1]) + sqr(dtemp[2,i]-dtemp[2,i-1])
           + sqr(dtemp[3,i]-dtemp[3,i-1]));
      par^[i]:=isum/sum;
    end;
end;

{=========================================================================}

Procedure GetNewDat(istat:word);
var
  i,j,g,num : word;

begin
  if (modedata<>1) and (modedata<>3) then exit;

  for i:=1 to nwlmod do
  begin
    num:=(istat-1)*nwlmod+i;
    w:=newwl[i]*y[jlh_wl+orderl];
    rbasis(orderl,jlh_wl,w,y,hmg,mbasis);
    for j:=1 to 3 do dnew[j]^[num]:=0;
    for j:=1 to jlh_wl do
      for g:=1 to 3 do
        dnew[g]^[num]:=dnew[g]^[num]+b2d[g]^[(istat-1)*jlh_wl+j]*mbasis[1,j];
  end;
end;
```

```
Procedure Bfit2dmod;
var
  h,i,j : word;

begin
  if (modedata<>1) and (modedata<>3) then exit;
  for i:=1 to nwlmod do new(nfit2d[i]);
  knotpar(nwlmod,orderl,newwl,y);
  for i:=1 to jlh_sta do
  begin
    for j:=1 to nwlmod do
    begin
      w:=newwl[j]*y[nwlmod+orderl];
      basis(orderl,nwlmod,w,y,mbasis);
      for h:=1 to nwlmod do nfit2d[j]^[h]:=mbasis[1,h];
    end;
    for j:=1 to nwlmod do
      for h:=1 to 3 do nfit2d[j]^[nwlmod+h] := dnew[h]^[(i-1)*nwlmod+j];
    gauss2d(nwlmod,nwlmod,nwlmod,3);
    for j:=1 to nwlmod do
      for h:=1 to 3 do b2dnew[h]^[(i-1)*nwlmod+j]:=nfit2d[j]^[nwlmod+h];
  end;
  for i:=1 to nwlmod do dispose(nfit2d[i]);
end;

{================Bspline curve fitting============}
Procedure Bfit2d(mode:word);
var
  g,h,i,j : word;

begin
  mplusc:= jlh_wl+orderl;
  new(wpar);
  Paramw;
  for i:=1 to jlh_wl do hmg[i]:=1;
  for i:=1 to jlh_wl do new(nfit2d[i]);
  for i:=1 to jlh_sta do
  begin
    for g:=1 to 3 do new(polytemp[g]);
    for g:=1 to jlh_wl do
      for h:=1 to 3 do
        polytemp[h]^[g]:=tempd[h,(i-1)*jlh_wl+g];
    knotc(jlh_wl,orderl,polytemp,y);
    for g:=1 to 3 do dispose(polytemp[g]);
    for j:=1 to jlh_wl do
    begin
      w:= wpar^[(i-1)*jlh_wl+j]*y[mplusc];
      basis(orderl,jlh_wl,w,y,mbasis);
      for h:=1 to jlh_wl do nfit2d[j]^[h]:=mbasis[1,h];
    end;
    for j:=1 to jlh_wl do
      for h:=1 to 3 do nfit2d[j]^[jlh_wl+h] := tempd[h,(i-1)*jlh_wl+j];
    gauss2d(jlh_wl,jlh_wl,jlh_wl,3);
    for j:=1 to jlh_wl do
      for h:=1 to 3 do b2d[h]^[(i-1)*jlh_wl+j] := nfit2d[j]^[jlh_wl+h];
    if mode=1 then GetNewDat(i);
  end;
  if wpar=nil then dispose(wpar);
  for i:=1 to jlh_wl do dispose(nfit2d[i]);
end;

{=====================================}
```

```
Procedure Bfit2dwl(mode:word);
var
  g,h,i,j,max : word;
  temp,da,db : single;


  {-------------------------}
  Procedure Calcpar(iwl:word);
  var
    i,j,n,nm,n1,n2 : word;
    marg : single;

  begin
    nstatss[iwl]:=jlh_sta;
    if (ssx[1,iwl]=0)and(ssx[2,iwl]=0) then
    begin
      for i:=1 to jlh_sta do
      for j:=1 to 3 do
      dtemp[j,i]:=tempd[j,(i-1)*jlh_WL+iwl];
    end
    else begin
    n1:=2;
    n2:=jlh_sta;
    if ssx[1,iwl]<0 then nstatss[iwl]:=nstatss[iwl]+1
      else
      begin
        marg:=tempd[1,iwl]+ssx[1,iwl];
        nm:=1;
        while round((stat^[nm]-shift)*1000)<=round(marg*1000) do inc(nm);
        nstatss[iwl]:=nstatss[iwl]-(nm-2);
        n1:=nm;
      end;
    if ssx[2,iwl]>0 then nstatss[iwl]:=nstatss[iwl]+1
      else
      begin
        marg:=tempd[1,(jlh_sta-1)*jlh_WL+iwl]+ssx[2,iwl];
        nm:=jlh_sta;
        while round((stat^[nm]-shift)*1000)>=round(marg*1000) do dec(nm);
        nstatss[iwl]:=nstatss[iwl]-(jlh_sta-nm-1);
        n2:=nm;
      end;

    if iwl=jlh_WL then nstatss[iwl]:=nstatss[iwl]+2;

    if ssx[1,iwl]<0 then
    begin
      dtemp[1,1]:=tempd[1,iwl]+ssx[1,iwl];
      if iwl=1 then dtemp[2,1]:=tempd[2,1] else dtemp[2,1]:=da;
      if iwl=jlh_WL then
      begin
        dtemp[1,2]:=dtemp[1,1];
        dtemp[2,2]:=0.01*tempd[2,round(jlh_sta/2)*jlh_WL];
        for i:=1 to 2 do dtemp[i,3]:=tempd[i,iwl];
        n:=3;
        if wsh=1 then
        begin
          for i:=1 to 2 do dtemp[3,i]:=ssz[1];
          dtemp[3,3]:=tempd[3,iwl];
        end;
      end
      else
      begin
        for i:=1 to 2 do dtemp[i,2]:=tempd[i,iwl];
        n:=2;
      end;
```

```
    end
    else
    begin
      dtemp[1,1]:=tempd[1,iwl]+ssx[1,iwl];dtemp[2,1]:=tempd[2,iwl];
      n:=1;
      if iwl=jlh_WL then
      begin
        dtemp[1,2]:=dtemp[1,1];
        dtemp[2,2]:=0.01*tempd[2,round(jlh_sta/2)*jlh_WL];
        for i:=1 to 2 do dtemp[i,3]:=tempd[i,iwl];
        n:=3;
        if wsh=1 then
        begin
          for i:=1 to 2 do dtemp[3,i]:=ssz[1];
          dtemp[3,3]:=tempd[3,iwl];
        end;
      end
    end;

    for i:=n1 to n2 do
    begin
      n:=n+1;
      for j:=1 to 3 do dtemp[j,n]:= tempd[j,(i-1)*jlh_WL+iwl];
    end;

{     if ssx[2,iwl]>0 then}
      if iwl=jlh_WL then
      begin
        dtemp[1,n+1]:=tempd[1,(jlh_sta-1)*jlh_WL+iwl]+ssx[2,iwl];
        dtemp[2,n+1]:=0.01*tempd[2,round(jlh_sta/2)*jlh_WL];
        dtemp[1,n+2]:=dtemp[1,n+1];
        dtemp[2,n+2]:=db;
        if wsh=1 then
          for i:=1 to 2 do dtemp[3,n+i]:=ssz[2];
      end
      else
      begin
        dtemp[1,n+1]:=tempd[1,(jlh_sta-1)*jlh_WL+iwl]+ssx[2,iwl];
        if iwl=1 then dtemp[2,n+1]:=tempd[2,(jlh_sta-1)*jlh_WL+1] else dtemp[2,n+1]:=db;
      end;
    end;
    if (wsh=0) or (iwl<>jlh_WL) then
      for i:=1 to nstatss[iwl] do dtemp[3,i]:=tempd[3,iwl];
    new(par);
    Paruperwl(nstatss[iwl]);
  end;

  {------------------------------}
  Procedure GetNewDatwl(iwl:word);
  var
    i,j,g,num : word;

  begin
    if modedata<>1 then exit;

    for i:=1 to nstatmod do
    begin
      num:=(i-1)*jlh_WL+iwl;
      u:=newstat[i]*xwl[iwl]^[nstatss[iwl]+orderk];
      basis(orderk,nstatss[iwl],u,xwl[iwl]^,nbasis);
      for j:=1 to 2 do dnew[j]^[num]:=0;
      for j:=1 to nstatss[iwl] do
        for g:=1 to 2 do
          dnew[g]^[num]:=dnew[g]^[num]+b2d[g]^[(j-1)*jlh_WL+iwl]*nbasis[1,j];
      dnew[3]^[num]:=tempd[3,iwl];
```

```
    end;
  end;

  {------------------------------}
  Procedure WithSheer;
  var
    g,num,h,i,j :word;
    u : single;

  begin
    for j:=1 to nstatss[jlh_WL] do
    begin
      writeln((jlh_WL-1)*nstatss[jlh_WL]+j);
      u:= par^[j]*xwl[jlh_WL]^[nstatss[jlh_WL]+orderk];
      basis(orderk,nstatss[jlh_WL],u,xwl[jlh_WL]^,nbasis);
      for h:=1 to nstatss[jlh_WL] do
        nfit2d[j]^[h] := nbasis[1,h];
    end;

    for j:=1 to nstatss[jlh_WL] do
    begin
      nfit2d[j]^[nstatss[jlh_WL]+1] := dtemp[1,j];
      nfit2d[j]^[nstatss[jlh_WL]+2] := dtemp[3,j];
    end;

    gauss2d(nstatss[jlh_WL],nstatss[jlh_WL],nstatss[jlh_WL],2);

    for j:=1 to nstatss[jlh_WL] do
      for h:=1 to 2 do b2d[h]^[nstatss[jlh_WL]*jlh_WL+j] :=
 nfit2d[j]^[nstatss[jlh_WL]+h];

    if modedata<>1 then exit;

    for i:=1 to nstatmod do
    begin
      num:=nstatmod*jlh_WL+i;
      u:=newstat[i]*xwl[jlh_WL]^[nstatss[jlh_WL]+orderk];
      basis(orderk,nstatss[jlh_WL],u,xwl[jlh_WL]^,nbasis);
      for j:=1 to 2 do dnew[j]^[num]:=0;
      for j:=1 to nstatss[jlh_WL] do
        for g:=1 to 2 do
          dnew[g]^[num]:=dnew[g]^[num]+b2d[g]^[nstatss[jlh_WL]*jlh_WL+j]*nbasis[1,j];
    end;
    for i:=1 to nstatmod do
      dnew[3]^[i*jlh_WL]:=dnew[2]^[nstatmod*jlh_WL+i];
  end;
  {------------------------------}


begin
  shift:=ssx[1,1];
  for i:=2 to jlh_WL do
    if ssx[1,i]<shift then shift:=ssx[1,i];

  for i:=1 to jlh_sta*jlh_WL do tempd[1,i]:=tempd[1,i]-shift;


  for i:=1 to jlh_sta+4 do new(nfit2d[i]);
  for g:=1 to 3 do new(polytemp[g]);


  for i:=1 to jlh_WL do
  begin

    Calcpar(i);
```

```
    if i=1 then begin da:=dtemp[2,1];db:=dtemp[2,nstatss[1]]; end;

    for g:=1 to nstatss[i] do
    begin
      for h:=1 to 2 do
        polytemp[h]^[g]:=dtemp[h,g];
      polytemp[3]^[g]:=tempd[3,i];
    end;
    knotc(nstatss[i],orderk,polytemp,xwl[i]^);
{    openknot(nstatss[i],orderk,xwl[i]^);}
    for j:=1 to nstatss[i] do
    begin
      u:= par^[j]*xwl[i]^[nstatss[i]+orderk];
      basis(orderk,nstatss[i],u,xwl[i]^,nbasis);
      for h:=1 to nstatss[i] do
        nfit2d[j]^[h] := nbasis[1,h];
    end;

    for j:=1 to nstatss[i] do
      for h:=1 to 2 do nfit2d[j]^[nstatss[i]+h]:=dtemp[h,j];

    gauss2d(nstatss[i],nstatss[i],nstatss[i],2);

    for j:=1 to nstatss[i] do
      for h:=1 to 2 do b2d[h]^[(j-1)*jlh_WL+i] := nfit2d[j]^[nstatss[i]+h];

    if mode=1 then GetNewDatwl(i);
    if (wsh=1) and (i=jlh_WL) then withSheer;

  end;
  dispose(par);
  for i:=1 to jlh_sta+2 do dispose(nfit2d[i]);
  for g:=1 to 3 do dispose(polytemp[g]);
end;

{=======================================================================}
Procedure Bfit2dmodwl;
var
  h,i,j : word;

begin
  if modedata<>1 then exit;
  for i:=1 to nstatmod do new(nfit2d[i]);
{  Openknot(nstatmod,k,x);}
  knotpar(nstatmod,orderk,newstat,x);
  for i:=1 to jlh_wl do
  begin
    for j:=1 to nstatmod do
    begin
      u:=newstat[j]*x[nstatmod+orderk];
      basis(orderk,nstatmod,u,x,nbasis);
      for h:=1 to nstatmod do nfit2d[j]^[h]:=nbasis[1,h];
    end;
    for j:=1 to nstatmod do
    for h:=1 to 2 do nfit2d[j]^[nstatmod+h] := dnew[h]^[(j-1)*jlh_wl+i];
    gauss2d(nstatmod,nstatmod,nstatmod,2);
    for j:=1 to nstatmod do
    for h:=1 to 2 do b2dnew[h]^[(j-1)*jlh_wl+i]:=nfit2d[j]^[nstatmod+h];
  end;
  for i:=1 to nstatmod do dispose(nfit2d[i]);
end;

{===========main program of Bspline==========}
Procedure Bsplbody(istat,num,po,mode:word);
var
```

```
    g,i,s : word;
    wplus,w : real;

begin
    w:=0;
    wplus:=y[num+orderl]/(po-1);
    for g:=1 to po do
    begin
      basis(orderl,num,w,y,mbasis);
      p2d[1]^[g].par:=0;
      p2d[2]^[g].par:=w;
      for s:=1 to 2 do p2d[s]^[g].pl:=0;
      for i:=1 to num do
        for s:=1 to 2 do
          if mode=1 then
 p2d[s]^[g].pl:=p2d[s]^[g].pl+b2d[s+1]^[(istat-1)*num+i]*mbasis[1,i]
            else p2d[s]^[g].pl:=p2d[s]^[g].pl+b2dnew[s+1]^[(istat-1)*num+i]*mbasis[1,i];
      w:= wplus*g;
    end;
end;

end.
```

```
1: unit Vdata3d;
2:
3: interface
4:
5: uses
6:   Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
7:   StdCtrls, ExtCtrls, Buttons, Menus,ta_decl,v3dproc,printers;
8:
9: type
10:   TFormview = class(TForm)
11:     MainMenu1: TMainMenu;
12:     Application1: TMenuItem;
13:     Exit1: TMenuItem;
14:     View1: TMenuItem;
15:     Mode1: TMenuItem;
16:     OrthogonalView1: TMenuItem;
17:     PersfektifView1: TMenuItem;
18:     GroupBox1: TGroupBox;
19:     BitBtnturnr: TBitBtn;
20:     BitBtnturnl: TBitBtn;
21:     BitBtnturnup: TBitBtn;
22:     BitBtndown: TBitBtn;
23:     BitBtnup: TBitBtn;
24:     BitBtnturnd: TBitBtn;
25:     BitBtnright: TBitBtn;
26:     BitBtnleft: TBitBtn;
27:     BitBtnbesar: TBitBtn;
28:     Bitbtnkecil: TBitBtn;
29:     Label1: TLabel;
30:     Edit1: TEdit;
31:     Edit2: TEdit;
32:     Label2: TLabel;
33:     Timer1: TTimer;
34:     BitBtn1: TBitBtn;
35:     Gaussmenu: TMenuItem;
36:     N1: TMenuItem;
37:     BodyPlan1: TMenuItem;
38:     WaterLines1: TMenuItem;
39:     ButtockLines1: TMenuItem;
40:     N2: TMenuItem;
41:     Print1: TMenuItem;
42:     N3: TMenuItem;
43:     Polygon1: TMenuItem;
44:     Options1: TMenuItem;
45:     Select1: TMenuItem;
46:     ChangePolygon1: TMenuItem;
47:     procedure Exit1Click(Sender: TObject);
48:     procedure BitBtnturnrMouseDown(Sender: TObject; Button: TMouseButton;
49:       Shift: TShiftState; X, Y: Integer);
50:     procedure BitBtnturnlMouseDown(Sender: TObject; Button: TMouseButton;
51:       Shift: TShiftState; X, Y: Integer);
52:     procedure BitBtnturnupMouseDown(Sender: TObject; Button: TMouseButton;
53:       Shift: TShiftState; X, Y: Integer);
54:     procedure BitBtnturndMouseDown(Sender: TObject; Button: TMouseButton;
55:       Shift: TShiftState; X, Y: Integer);
56:     procedure BitBtndownMouseDown(Sender: TObject; Button: TMouseButton;
57:       Shift: TShiftState; X, Y: Integer);
58:     procedure BitBtnupMouseDown(Sender: TObject; Button: TMouseButton;
59:       Shift: TShiftState; X, Y: Integer);
60:     procedure BitBtnrightMouseDown(Sender: TObject; Button: TMouseButton;
61:       Shift: TShiftState; X, Y: Integer);
62:     procedure BitBtnleftMouseDown(Sender: TObject; Button: TMouseButton;
63:       Shift: TShiftState; X, Y: Integer);
64:     procedure BitBtnbesarMouseDown(Sender: TObject; Button: TMouseButton;
65:       Shift: TShiftState; X, Y: Integer);
```

```
    procedure BitbtnkecilMouseDown(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure Timer1Timer(Sender: TObject);
    procedure BitBtnturnrMouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure BitBtnturnlMouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure BitBtnturnupMouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure BitBtnturndMouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure BitBtndownMouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure BitBtnupMouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure BitBtnrightClick(Sender: TObject);
    procedure BitBtnleftClick(Sender: TObject);
    procedure BitBtnbesarClick(Sender: TObject);
    procedure Bitbtnkecilclick(Sender: TObject);
    procedure OrthogonalView1Click(Sender: TObject);
    procedure PersfektifView1Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure GaussmenuClick(Sender: TObject);
    procedure BodyPlan1Click(Sender: TObject);
    procedure WaterLines1Click(Sender: TObject);
    procedure ButtockLines1Click(Sender: TObject);
    procedure FormDblClick(Sender: TObject);
    procedure Print1Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure Edit1Exit(Sender: TObject);
    procedure Edit1KeyPress(Sender: TObject; var Key: Char);
    procedure Polygon1Click(Sender: TObject);
    procedure Select1Click(Sender: TObject);
    procedure FormMouseDown(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure ChangePolygon1Click(Sender: TObject);
    procedure Edit2Exit(Sender: TObject);
    procedure Edit2KeyPress(Sender: TObject; var Key: Char);
    private
    Procedure DrawHB(mode:word);
    procedure DrawHull(mode:word);
    Procedure Floor(mode:word);
    Procedure showangle;
    Procedure DrawBPpar;
    Procedure DrawLine(mode:word);
    Procedure PrintLine(mode:word);
    { Private declarations }
    public
    Bufimage:Tbitmap;
    { Public declarations }
  end;
  procedure origin;
  Procedure convert(mode:word;ex,ey,ez:single ;var xs,ys:single);


var
  Formview: TFormview;

implementation

uses gaussclr, PLchange;

var
  xmax,ymax:single;
  besar,kecil,kiri,kanan,atas,bawah,
```

```
  putarki,putarka,putara,putarb,tukarmode       :boolean;
  polyvar: array [1..4] of Tpoint;
{$R *.DFM}

{==========Return surface coordinates to origin axis==========}

Procedure Origin;
var i : word;
begin
  if center then
  begin
    for i:=1 to po2*po1 do
    begin
      p[1]^[i].pl:=p[1]^[i].pl+xcen;
{      p[3]^[i].pl:=p[3]^[i].pl+zcen;}
    end;
    center:=false;
  end;
end;


procedure TFormview.Exit1Click(Sender: TObject);
var i:integer;
begin
  for i:=1 to 3 do
  begin
    if bsc[i]<>nil then dispose(bsc[i]);
    if sc[i]<>nil then dispose(sc[i]);
  end;
  for i:=1 to 2 do if sc2[i]<>nil then dispose(sc2[i]);
  bufimage.free;
  if showwhat=4 then
  begin
    gaussform.close;
    gaussform.free;
    formCP.close;
    formCP.free;
  end;
  close;
end;

{==================First Value==================}
Procedure FirstValue;
var i,j :word;
begin
  for i:=1 to 7 do
  begin
    gaussbts[i]:=(7-i)*rentangwarnaneg/6+gaussmaxneg;
    gaussbts[i+7]:=(i-1)*rentangwarnapos/6+gaussminpos;
  end;
  gaussbts[7]:=0;gaussbts[8]:=0;
  warnaf[0]:=RGB(0,150,150);
  warnaf[1]:=RGB(0,0,125);
  warnaf[2]:=RGB(0,0,255);
  warnaf[3]:=RGB(0,175,255);
  warnaf[4]:=RGB(0,255,255);
  warnaf[5]:=RGB(255,255,255);
  warnaf[6]:=RGB(255,125,200);
  warnaf[7]:=RGB(255,150,50);
  warnaf[8]:=RGB(255,0,0);
  warnaf[9]:=RGB(175,0,90);
  warnaf[10]:=RGB(150,0,0);
  dis:=22.5; vd:=1; s:=3;
  scf:=1.1049; theta0:=270; phi0:=90;
  vx:=100; vy:=100; cy:=-1; cx:=1.9;
```

```
   center:=false;  HBcenter:=false; Lcenter :=false;
   full:=true; showbd:=true;showwl:=true; showbl:=true;
   updateview:=false;viewgausscurv:=false;viewpolyvert:=false;
   selectcondition:=false;changepolystatus:=false;
   diss:=dis; vds:=vd; ss:=s;
   thetas:=theta0;phis:=phi0;
   vxs:=vx; vys:=vy; cys:=cy;cxs:=cx;

   Lthet:=0; Lphi:=0; Ldis:=5;rendnum:=1;
   kd:=1; ks:=1; Ia:=0.1; Ip:=1; d0:=1; shin:=100;
   theta :=theta0*pi/180 ; phi := phi0*pi/180;
   sn1:=sin(theta);  cn1:=cos(theta);
   sn2:=sin(phi);    cn2:=cos(phi);

   for i:=1 to 3 do
   begin
     new(bsc[i]);
     new(sc[i]);
     for j:=1 to maxdpts do
     begin
       bsc[i]^[j]:=0;
       sc[i]^[j]:=0;
     end;
   end;
   for i:=1 to 2 do new(sc2[i]);
end;

{============== Convert the 3D coordinat to 2D screen coordinat=========}
Procedure convert(mode:word;ex,ey,ez:single ;var xs,ys:single);
var
  xe,ye,ze : single;

begin
    xe:=-ex*sn1+ey*cn1;
    ye:=-ex*cn1*cn2-ey*sn1*cn2+ez*sn2;
    ze:=-ex*sn2*cn1-ey*sn1*sn2-ez*cn2+dis;

    if (s>0) and (ze>0) then
    begin
      if mode=2 then
      begin
        xs:=(s/vd)*(xe/ze);
        ys:=(s/vd)*(ye/ze);
      end
      else
      begin
        xs:=(s/vd)*(xe/dis);
        ys:=(s/vd)*(ye/dis);
      end;
      xs:=scf*((xs+1+cx)*vx);
      ys:=(-ys+1-cy)*vy;
    end;
end;


{==================Draw Floor=====================}
Procedure Tformview.Floor(mode:word);
var
  xpix,ypix: integer;
  axis  : array[1..2,1..2] of single;
  floorcor : array[1..2] of single;
  xscl,yscl : single;

begin
  if xmax/2>dis then dis:=1.5*xmax/2;
```

```
  xscl:=xmax*0.75/15;
  yscl:=ymax*3/15;
  bufimage.canvas.pen.color:=clgreen;
    for xpix:=-15 to 15 do
      for ypix:=-15 to 15 do
      begin
        convert(mode,xpix*xscl,ypix*yscl,0,floorcor[1],floorcor[2]);
        bufimage.canvas.pixels[round(floorcor[1]),round(floorcor[2])]:=clgreen;
      end;
   bufimage.canvas.pen.Color:=clgreen;
   bufimage.canvas.pen.style:=psDashDot;
   convert(mode,-15*xscl,0,0,axis[1,1],axis[1,2]);
   convert(mode,15*xscl,0,0,axis[2,1],axis[2,2]);
   bufimage.canvas.moveto(round(axis[1,1]),round(axis[1,2]));
   bufimage.canvas.lineto(round(axis[2,1]),round(axis[2,2]));
   convert(mode,0,-15*yscl,0,axis[1,1],axis[1,2]);
   convert(mode,0,15*yscl,0,axis[2,1],axis[2,2]);
   bufimage.canvas.moveto(round(axis[1,1]),round(axis[1,2]));
   bufimage.canvas.lineto(round(axis[2,1]),round(axis[2,2]));
   bufimage.canvas.pen.style:=pssolid;
end;

{=========================================================}
Procedure Tformview.showangle;
begin
  tilt:=phi0-90;
  edit1.text:=floattostr(theta0);
  edit2.text:=floattostr(tilt);
end;


{==================Drawing ship hull=====================}
Procedure Tformview.DrawHull(mode:word);
var
  i,j,num : word;
  X1,Y1 : longint;


begin
  if showdat then
  for i:=1 to jlh_sta*jlh_WL do
  convert(mode,tempd[1,i],tempd[2,i],tempd[3,i],bsc[1]^[i],bsc[2]^[i]);
  for j:=1 to po1*po2 do
  convert(mode,p[1]^[j].pl,p[2]^[j].pl,p[3]^[j].pl,sc[1]^[j],sc[2]^[j]);
  if full then
  for j:=1 to po1*po2 do
  convert(mode,p[1]^[j].pl,-p[2]^[j].pl,p[3]^[j].pl,sc2[1]^[j],sc2[2]^[j]);

  Floor(mode);
  showangle;

{  if showdat then for i:=1 to jlh sta*jlh WL do
   cross(round(bsc[1]^[i]),round(bsc[2]^[i]),yellow);
}
  if showbd then
  for i:=1 to po1 do
  begin
    if (i=1) or (i=po1) then bufimage.canvas.pen.color:=clwhite
    else bufimage.canvas.pen.color:=clblue;
    num:=(i-1)*po2+1;
    x1:=round(sc[1]^[num]);
    y1:=round(sc[2]^[num]);
    bufimage.canvas.moveto(x1,y1);
    for j:=2 to po2 do
    begin
```

```
      x1:=round(sc[1]^[num-1+j]);
      y1:=round(sc[2]^[num-1+j]);
      bufimage.canvas.lineto (x1,y1);
    end;
    if full then
    begin
      x1:=round(sc[1]^[num]);
      y1:=round(sc[2]^[num]);
      bufimage.canvas.moveto(x1,y1);
      for j:=1 to po2 do
      begin
        x1:=round(sc2[1]^[num-1+j]);
        y1:=round(sc2[2]^[num-1+j]);
        bufimage.canvas.lineto (x1,y1);
      end;
    end;
  end;{for po1}

  if showwl then
  for i:=1 to po2 do
  begin
    if i=po2 then bufimage.canvas.pen.color:=Tcolor(RGB(255,255,255))
    else bufimage.canvas.pen.color:=Tcolor(RGB(255,0,0));
    x1:=round(sc[1]^[i]);
    y1:=round(sc[2]^[i]);
    bufimage.canvas.moveto(x1,y1);
    for j:=2 to po1 do
    begin
      num:=(j-1)*po2+i;
      x1:=round(sc[1]^[num]);
      y1:=round(sc[2]^[num]);
      bufimage.canvas.lineto(x1,y1);
    end;
    if full then
    begin
      x1:=round(sc2[1]^[i]);
      y1:=round(sc2[2]^[i]);
      bufimage.canvas.moveto(x1,y1);
      for j:=2 to po1 do
      begin
        num:=(j-1)*po2+i;
        x1:=round(sc2[1]^[num]);
        y1:=round(sc2[2]^[num]);
        bufimage.canvas.lineto(x1,y1);
      end;
    end;
  end;{for jlh WL}
end;


{==================Drawing ship hull half breadth=====================}
Procedure Tformview.DrawHB(mode:word);
var
  i,j : word;
  num:integer;

begin
  for j:=1 to jlh_sta*jlh_wl do
  convert(mode,tempd[1,j],tempd[2,j],tempd[3,j],sc[1]^[j],sc[2]^[j]);
  if full then for j:=1 to
  jlh_sta*jlh_wl do
  convert(mode,tempd[1,j],-tempd[2,j],tempd[3,j],sc2[1]^[j],sc2[2]^[j]);
  Floor(mode);
  showangle;
```

```
  if showbd then
  begin
    for i:=1 to jlh_sta do
    begin

      if (i=1) or (i=jlh_sta) then bufimage.canvas.pen.color:=clwhite
      else bufimage.canvas.pen.color:=clblue;
      num:=(i-1)*jlh_wl+1;
      x1:=round(sc[1]^[num]);
      y1:=round(sc[2]^[num]);
      bufimage.canvas.moveto(x1,y1);
      for j:=2 to jlh_wl do
      begin
        x1:=0;y1:=0;
        x1:=round(sc[1]^[num-1+j]);
        y1:=round(sc[2][num-1+j]);
        bufimage.canvas.lineto (x1,y1);
      end;
      if full then
      begin
        x1:=round(sc[1]^[num]);
        y1:=round(sc[2]^[num]);
        bufimage.canvas.moveto(x1,y1);
        for j:=1 to jlh_wl do
        begin
          x2:=0;y2:=0;
          x2:=round(sc2[1]^[num-1+j]);
          y2:=round(sc2[2]^[num-1+j]);
          bufimage.canvas.lineto (x2,y2);
        end;
      end;
    end;{for po1}
  end;
  if showwl then
  begin
    for i:=1 to jlh_wl do
    begin
      if i=jlh_wl then bufimage.canvas.pen.color:=Tcolor(RGB(255,255,255))
      else bufimage.canvas.pen.color:=Tcolor(RGB(255,0,0));
      x1:=round(sc[1]^[i]);
      y1:=round(sc[2]^[i]);
      bufimage.canvas.moveto(x1,y1);
      for j:=2 to jlh_sta do
      begin
        num:=(j-1)*jlh_wl+i;
        x1:=round(sc[1]^[num]);
        y1:=round(sc[2]^[num]);
        bufimage.canvas.lineto(x1,y1);
      end;
      if full then
      begin
        x1:=round(sc2[1]^[i]);
        y1:=round(sc2[2]^[i]);
        bufimage.canvas.moveto(x1,y1);
        for j:=2 to jlh_sta do
        begin
          num:=(j-1)*jlh_wl+i;
          x2:=round(sc2[1]^[num]);
          y2:=round(sc2[2]^[num]);
          bufimage.canvas.lineto(x2,y2);
        end;
      end;
    end;{for jlh_wl }
  end;
end;
```

```
{================Draw Parametric Body Plan 2D===============}
Procedure Tformview.DrawBPpar;
var
  h,i,j,ypos,zpos,i1 :word;
  p2,p3 : real;


begin
  scale:=200;
  y0:=320;z0:=450-round((400-(1.5*scale))*0.5);
  bufimage.canvas.pen.color:=clred;
  bufimage.canvas.pen.style:=psdashdot;
  bufimage.canvas.moveto(y0,10);
  bufimage.canvas.lineto(y0,450);
  bufimage.canvas.pen.style:=pssolid;
  bufimage.canvas.pen.color:=clwhite;
  bufimage.canvas.moveto(y0,z0);

  for i:=1 to po1 do
    for j:=1 to po2 do
    begin
      p2:=p[2]^[(i-1)*po2+j].pl * scale;
      p3:=p[3]^[(i-1)*po2+j].pl * scale;
      if i <= (po1/2) then ypos:=y0-round(p2)
        else ypos:=round(p2)+y0;
      zpos:=z0-round(p3);
      if j=1 then bufimage.canvas.moveto(ypos,zpos)
      else bufimage.canvas.lineto(ypos,zpos);
    end;


  for j:=1 to po2 do
  begin
    h:=0;
    for i:=1 to po1 do
    begin
      p2:=p[2]^[(i-1)*po2+j].pl * scale;
      p3:=p[3]^[(i-1)*po2+j].pl * scale;
      zpos:=z0-round(p3);
      if i <= (po1/2) then
      begin
        ypos:=y0-round(p2);
        if h<>1 then i1:=1 else i1:=0;
        h:=1;
      end
      else
      begin
        ypos:=round(p2)+y0;
        if h<>2 then i1:=1 else i1:=0;
        h:=2;
      end;

      if i1=1 then bufimage.canvas.moveto(ypos,zpos)
      else bufimage.canvas.lineto(ypos,zpos);
    end;
  end;
end;


{================Drawing Lines=====================}
Procedure Tformview.DrawLine(mode:word);
var
  i,j,num,num2 : integer;
  X1,Y1,x2,y2 : longint;
```

```
    denum,xsccen,ysccen: single;
    grad,cons  : array[1..3] of single;
begin
    timer1.enabled:=false;
    Floor(mode);
    showangle;
    if viewgausscurv then
    begin
      for j:=1 to gnum*gprt do
  convert(mode,gvcor^[j].x,gvcor^[j].y,gvcor^[j].z,sc[1]^[j],sc[2]^[j]);
      for j:=1 to (gnum)*(gprt)do
      begin
        if gvcor^[j].warna<=0 then
        begin
          if gvcor^[j].warna<=-1 then warna[j]:=0
          else begin
            if gvcor^[j].warna<=-0.1 then warna[j]:=1
            else begin
              if gvcor^[j].warna<=-0.01 then warna[j]:=2
              else begin
                if gvcor^[j].warna<=-0.001 then warna[j]:=3
                else begin
                  if gvcor^[j].warna<=-0.0001 then warna[j]:=4
                  else warna[j]:=5;
                end;
              end;
            end;
          end;
        end
        else begin
          if gvcor^[j].warna>=1 then warna[j]:=10
          else begin
            if gvcor^[j].warna>=0.1 then warna[j]:=9
            else begin
              if gvcor^[j].warna>=0.01 then warna[j]:=8
              else begin
                if gvcor^[j].warna>=0.001 then warna[j]:=7
                else begin
                  if gvcor^[j].warna>=0.0001 then warna[j]:=6
                  else warna[j]:=5;
                end;
              end;
            end;
          end;
        end;
      end;
      bufimage.canvas.pen.Width:=1;
      bufimage.canvas.Brush.style:=bssolid;
      for i:=1 to gnum-1 do
      begin
        for j:=1 to gprt -1do
        begin
          num:=(i-1)*gprt+j;
          num2:=(i)*gprt+j;
          polyvar[1].x:=round(sc[1]^[num]);
          polyvar[1].y:=round(sc[2]^[num]);
          polyvar[2].x:=round(sc[1]^[num2]);
          polyvar[2].y:=round(sc[2]^[num2]);
          polyvar[4].x:=round(sc[1]^[num+1]);
          polyvar[4].y:=round(sc[2]^[num+1]);
          polyvar[3].x:=round(sc[1]^[num2+1]);
          polyvar[3].y:=round(sc[2]^[num2+1]);
          if (num=1)then
          begin
            bufimage.canvas.pen.color:=warnaf[warna[num]];
```

```
            bufimage.canvas.Brush.Color:=warnaf[warna[num]];
            color_use:=warnaf[warna[num]];
          end
          else if (warnaf[warna[num]]<>color_use)then
          begin
            bufimage.canvas.pen.color:=warnaf[warna[num]];
            bufimage.canvas.Brush.Color:=warnaf[warna[num]];
            color_use:=warnaf[warna[num]];
          end;
          bufimage.canvas.polygon(slice(polyvar,4));
        end;
      end;
    end;
    for j:=1 to 2*bdprt do
  convert(mode,sscor[1]^[j],sscor[2]^[j],sscor[3]^[j],sc[1]^[j],sc[2]^[j]);
    bufimage.canvas.pen.color:=clwhite;
    x1:=round(sc[1]^[1]);
    y1:=round(sc[2]^[1]);
    bufimage.canvas.moveto(x1,y1);
    for j:=2 to bdprt do
    begin
      x1:=round(sc[1]^[j]);
      y1:=round(sc[2]^[j]);
      bufimage.canvas.lineto (x1,y1);
    end;
    x1:=round(sc[1]^[1]);
    y1:=round(sc[2]^[1]);
    bufimage.canvas.moveto(x1,y1);
    for j:=1 to bdprt do
    begin
      x1:=round(sc[1]^[bdprt+j]);
      y1:=round(sc[2]^[bdprt+j]);
      bufimage.canvas.lineto (x1,y1);
    end;

    if showbd then
    begin
      for j:=1 to bdnum*bdprt do
  convert(mode,bdcor[1]^[j],bdcor[2]^[j],bdcor[3]^[j],sc[1]^[j],sc[2]^[j]);
      if full then
        for j:=1 to bdnum*bdprt do
  convert(mode,bdcor[1]^[j],-bdcor[2]^[j],bdcor[3]^[j],sc2[1]^[j],sc2[2]^[j]);
      bufimage.canvas.pen.color:=clyellow;
      for i:=1 to bdnum do
      begin
        num:=(i-1)*bdprt+1;
        x1:=round(sc[1]^[num]);
        y1:=round(sc[2]^[num]);
        bufimage.canvas.  moveto(x1,y1);
        for j:=2 to bdprt do
        begin
          x1:=round(sc[1]^[num-1+j]);
          y1:=round(sc[2]^[num-1+j]);
          bufimage.canvas.lineto(x1,y1);
        end;
        if full then
        begin
          x1:=round(sc[1]^[num]);
          y1:=round(sc[2]^[num]);
          bufimage.canvas.moveto(x1,y1);
          for j:=1 to bdprt do
          begin
            x2:=round(sc2[1]^[num-1+j]);
            y2:=round(sc2[2]^[num-1+j]);
            bufimage.canvas.lineto (x2,y2);
```

```
      end;
    end;
  end;{for pol}
end;

if showwl then
begin
  for j:=1 to wlprt*wlnum do
convert(mode,wlcor[1]^[j],wlcor[2]^[j],wlcor[3]^[j],sc[1]^[j],sc[2]^[j]);
  if full then
    for j:=1 to wlprt*wlnum do
convert(mode,wlcor[1]^[j],-wlcor[2]^[j],wlcor[3]^[j],sc2[1]^[j],sc2[2]^[j]);

  bufimage.canvas.pen.color:=clred;
  for i:=1 to wlnum do
  begin
    if i=wlnum then bufimage.canvas.pen.color:=clwhite;
    num:=(i-1)*wlprt+1;
    x1:=round(sc[1]^[num]);
    y1:=round(sc[2]^[num]);
    bufimage.canvas.moveto(x1,y1);
    for j:=2 to wlprt do
    begin
      x1:=round(sc[1]^[num-1+j]);
      y1:=round(sc[2]^[num-1+j]);
      bufimage.canvas.lineto(x1,y1);
    end;
    if full then
    begin
      x1:=round(sc[1]^[num]);
      y1:=round(sc[2]^[num]);
      bufimage.canvas.moveto(x1,y1);
      for j:=1 to wlprt do
      begin
        x2:=round(sc2[1]^[num-1+j]);
        y2:=round(sc2[2]^[num-1+j]);
        bufimage.canvas.lineto(x2,y2);
      end;
    end;
  end;{for nwl}
end;

if showbl then
begin
  for j:=1 to blprt*blnum do
onvert(mode,blcor[1]^[j],blcor[2]^[j],blcor[3]^[j],sc[1]^[j],sc[2]^[j]);
  if full then
    for j:=1 to blprt*blnum do
onvert(mode,blcor[1]^[j],-blcor[2]^[j],blcor[3]^[j],sc2[1]^[j],sc2[2]^[j]);

  bufimage.canvas.pen.color:=RGB(0,150,255);
  for i:=1 to blnum do
  begin
    num:=(i-1)*blprt+1;
    x1:=round(sc[1]^[num]);
    y1:=round(sc[2]^[num]);
    bufimage.canvas.moveto(x1,y1);
    for j:=2 to blprt do
    begin
      x1:=round(sc[1]^[num-1+j]);
      y1:=round(sc[2]^[num-1+j]);
      bufimage.canvas.lineto(x1,y1);
    end;
    if full then
    begin
```

```
      x1:=round(sc2[1]^[num]);
      y1:=round(sc2[2]^[num]);
      bufimage.canvas.moveto(x1,y1);
      for j:=1 to blprt do
      begin
        x2:=round(sc2[1]^[num-1+j]);
        y2:=round(sc2[2]^[num-1+j]);
        bufimage.canvas.lineto(x2,y2);
      end;
    end;
  end;
end;
if viewpolyvert then
begin
  bufimage.canvas.pen.Color:=clgreen;
  bufimage.canvas.pen.Width:=2;
  for j:=1 to jlh_WL*jlh_sta do
onvert(mode,b[1]^[j],b[2]^[j],b[3]^[j],sc[1]^[j],sc[2]^[j]);
  for i:=1 to jlh_sta-1 do
  begin
    num:=(i-1)*jlh_wl+1;
    num2:=i*jlh_wl+1;
    polyvar[1].x:=round(sc[1]^[num]);
    polyvar[1].y:=round(sc[2]^[num]);
    polyvar[2].x:=round(sc[1]^[num2]);
    polyvar[2].y:=round(sc[2]^[num2]);
    for j:=2 to jlh_WL do
    begin
      polyvar[4].x:=round(sc[1]^[num-1+j]);
      polyvar[4].y:=round(sc[2]^[num-1+j]);
      polyvar[3].x:=round(sc[1]^[num2-1+j]);
      polyvar[3].y:=round(sc[2]^[num2-1+j]);
      bufimage.canvas.moveto(polyvar[1].x,polyvar[1].y);
      bufimage.canvas.lineto(polyvar[2].x,polyvar[2].y);
      bufimage.canvas.lineto(polyvar[3].x,polyvar[3].y);
      bufimage.canvas.lineto(polyvar[4].x,polyvar[4].y);
      bufimage.canvas.lineto(polyvar[1].x,polyvar[1].y);
      polyvar[1].x:=polyvar[4].x;
      polyvar[1].y:=polyvar[4].y;
      polyvar[2].x:=polyvar[3].x;
      polyvar[2].y:=polyvar[3].y;
    end;
  end;
end;
bufimage.canvas.pen.Width:=1;
timer1.enabled:=true;
nd;


==========View Half Breadth==========}
rocedure TFormview.BitBtnturnrMouseDown(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
egin
putarka:=true;
nd;

rocedure TFormview.BitBtnturnlMouseDown(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
egin
putarki:=true;
nd;

rocedure TFormview.BitBtnturnupMouseDown(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
```

```
gin
putara:=true;
d;

ocedure TFormview.BitBtnturndMouseDown(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
gin
putarb:=true;
d;

ocedure TFormview.BitBtndownMouseDown(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
gin
bawah:=true;
d;

ocedure TFormview.BitBtnupMouseDown(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
gin
atas:=true;
d;

ocedure TFormview.BitBtnrightMouseDown(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
gin
kanan:=true;
d;

ocedure TFormview.BitBtnleftMouseDown(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
gin
kiri:=true;
d;

ocedure TFormview.BitBtnbesarMouseDown(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
gin
besar:=true;
d;

ocedure TFormview.Bitbtnkecilmousedown(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
gin
kecil:=true;
d;

ocedure TFormview.Timer1Timer(Sender: TObject);
r
R:TRect;
i:integer;
gin
if pertama then
begin
  bufimage:=Tbitmap.create;
  bufimage.width:=775;
  bufimage.height:=490;
  if showwhat=4 then
  begin
    gaussmenu.enabled:=true;
    polygon1.enabled:=true;
    gaussform:=Tgaussform.create(self);
    gaussform.formstyle:=fsStayOnTop;
    formCP:=TformCP.create(self);
    formCP.formStyle:=fsStayOnTop;
  end
```

```
  else begin
    gaussmenu.enabled:=false;
    polygon1.enabled:=false;
  end;


end;
case showwhat of
{HB}:
if  pertama then
begin
  orthogonalview1.enabled:=false;
  mode:=1;
  Firstvalue;
  if not HBcenter then
  begin
    xcen:=(0.5*(tempd[1,1]+tempd[1,jlh_WL*jlh_sta]));
    for i:=1 to jlh_WL*jlh_sta do tempd[1,i]:=tempd[1,i]-xcen;
    HBcenter := true;
  end;
  xmax:=abs(tempd[1,jlh_WL]-tempd[1,jlh_WL*jlh_sta]);
  ymax:=tempd[2,round(jlh_sta/2)*jlh_WL];
end;

{HULL}:
if pertama then
begin
  orthogonalview1.enabled:=false;
  mode:=1;
  Firstvalue;
  if not center then
  begin
    xcen:=(0.5*(p[1]^[1].pl+p[1]^[po1*po2].pl));
    for i:=1 to po1*po2 do
    begin
      p[1]^[i].pl:=p[1]^[i].pl-xcen;
    end;
    center := true;
  end;
  xmax:=abs(p[1]^[po2*po1].pl-p[1]^[po2].pl);
  ymax:=p[2]^[round(po1/2)*po2].pl;
end;
3:{BPPAR}
begin
  orthogonalview1.enabled:=false;
end;
4:{LINES}
begin
  if pertama then
  begin
    Bodyplan1.Checked:=true;
    Waterlines1.Checked:=true;
    Buttocklines1.Checked:=true;
    gaussmenu.checked:=false;
    polygon1.checked:=false;
    orthogonalview1.enabled:=false;
    mode:=1;
    Firstvalue;
    if not Lcenter then
    begin
      xcen:=(0.5*(sscor[1]^[bdprt]+sscor[1]^[2*bdprt]));
      for i:=1 to bdnum*bdprt do
       bdcor[1]^[i]:=bdcor[1]^[i]-xcen;
      for i:=1 to wlnum*wlprt do
       wlcor[1]^[i]:=wlcor[1]^[i]-xcen;
```

```
      for i:=1 to blnum*blprt do
       blcor[1]^[i]:=blcor[1]^[i]-xcen;
      for i:=1 to 2*bdprt do
       sscor[1]^[i]:=sscor[1]^[i]-xcen;
      for i:=1 to gnum*gprt do
       gvcor^[i].x:=gvcor^[i].x-xcen;
      for i:=1 to jlh_WL*jlh_sta do
       b[1]^[i]:=b[1]^[i]-xcen;
      Lcenter := true;
    end;
    xmax:=abs(sscor[1]^[bdprt]-sscor[1]^[2*bdprt]);
    ymax:=wlcor[2]^[(wlnum-1)*round(wlprt/2)+wlprt];
  end;
end;{LINES}
end;{CASE}

if (besar)or (kecil)or(pertama)or(kiri)or(kanan)or(atas)or(bawah)
  or(putarki)or(putarka)or(putara)or(putarb)or(tukarmode)or(updateview) then
begin
  if besar then s:=s*1.5;
  if kecil then s:=s/1.5;
  if kiri then cx:=cx-0.1;
  if kanan then cx:=cx+0.1;
  if atas then   cy:=cy+0.1;
  if bawah then cy:=cy-0.1;
  if putarki then
  begin
    theta0:=theta0+5;
    if theta0>=360 then theta0:=theta0-360;
  end;
  if putarka then
  begin
    theta0:=theta0-5;
    if theta0<=0 then theta0:=360+theta0;
  end;
  if putarb then
  begin
    phi0:=phi0-3;
    if (phi0)<=90 then phi0:=360+phi0;
  end;
  if putara then
  begin
    phi0:=phi0+3;
    if phi0>=450 then phi0:=phi0-360;
  end;
  R:=rect(0,0,775,500);
  theta:=theta0*pi/180 ; phi:=phi0*pi/180;
  sn1:=sin(theta);  cn1:=cos(theta);
  sn2:=sin(phi);    cn2:=cos(phi);
  bufimage.canvas.brush.style:=bssolid;
  bufimage.canvas.brush.color:=clblack;
  bufimage.canvas.rectangle(R.left,R.top,R.right,R.bottom);
  case showwhat of
  1{'HB'}: DrawHB(mode);
  2{'HULL'}:DrawHull(mode);
  3{BPPAR}:
  begin
     DRAWBPPAR;
     origin;
  end;
  4:{LINES}
  begin
    DrawLine(mode);
  end
  end;
```

```
  bitblt(canvas.handle,5,50,780,540,
     bufimage.canvas.handle,0,0,srccopy);
  pertama:=false;tukarmode:=false;updateview:=false;
end;
i;


ocedure TFormview.BitBtnturnrMouseUp(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
gin
putarka:=false;
d;

ocedure TFormview.BitBtnturnlMouseUp(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
gin
putarki:=false;
d;

ocedure TFormview.BitBtnturnupMouseUp(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
gin
putara:=false;
d;

ocedure TFormview.BitBtnturndMouseUp(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
gin
putarb:=false;
d;

ocedure TFormview.BitBtndownMouseUp(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
gin
bawah:=false;
d;

ocedure TFormview.BitBtnupMouseUp(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
gin
atas:=false;
d;

ocedure TFormview.BitBtnrightClick(Sender: TObject);
gin
kanan:=false;
d;

ocedure TFormview.BitBtnleftClick(Sender: TObject);
gin
kiri:=false;
d;

ocedure TFormview.BitBtnbesarClick(Sender: TObject);
gin
besar:=false;
d;

ocedure TFormview.BitbtnkecilClick(Sender: TObject);
gin
kecil:=false;
d;

ocedure TFormview.OrthogonalView1Click(Sender: TObject);
gin
```

```
persfektifview1.enabled:=true;
orthogonalview1.enabled:=false;
mode:=1;tukarmode:=true;
d;

ocedure TFormview.PersfektifView1Click(Sender: TObject);
gin
persfektifview1.enabled:=false;
orthogonalview1.enabled:=true;
mode:=2;
tukarmode:=true;
d;

ocedure TFormview.BitBtn1Click(Sender: TObject);
gin
if  full then
begin
  full:=false;
end
else begin
  full:=true;
end;
tukarmode:=true;
d;

ocedure TFormview.GaussmenuClick(Sender: TObject);
gin
if not(Gaussmenu.checked) then
begin
  Gaussmenu.checked:=true;
  Viewgausscurv:=true;
end
else begin
  Gaussmenu.checked:=false;
  viewgausscurv:=false;
end;
tukarmode:=true;
d;

ocedure TFormview.BodyPlan1Click(Sender: TObject);
gin
if not(Bodyplan1.Checked) then
begin
  Bodyplan1.checked:=true;
  showbd:=true;
end
else begin
  Bodyplan1.checked:=false;
  showbd:=false;
end;
tukarmode:=true;
d;

ocedure TFormview.WaterLines1Click(Sender: TObject);
gin
if not(Waterlines1.Checked) then
begin
  Waterlines1.checked:=true;
  showwl:=true;
end
else begin
  Waterlines1.checked:=false;
  showwl:=false;
end;
tukarmode:=true;
```

```
i;

ocedure TFormview.ButtockLines1Click(Sender: TObject);
gin
if not(Buttocklines1.Checked) then
egin
  Buttocklines1.checked:=true;
  showbl:=true;
end
else begin
  Buttocklines1.checked:=false;
  showbl:=false;
end;
tukarmode:=true;
d;

ocedure TFormview.FormDblClick(Sender: TObject);
gin
if Gaussmenu.checked then
begin
  gaussform.show;
 end;
d;


==========Print Lines==========}
ocedure Tformview.printLine(mode:word);
r
i,j,num,num2 : integer;
X1,Y1,x2,y2 : longint;
denum,xsccen,ysccen: single;
grad,cons  : array[1..3] of single;
polyvar: array [1..4] of Tpoint;
gin
Floor(mode);
showangle;
if viewgausscurv then
begin
  for j:=1 to gnum*gprt do
convert(mode,gvcor^[j].x,gvcor^[j].y,gvcor^[j].z,sc[1]^[j],sc[2]^[j]);
  for j:=1 to (gnum)*(gprt)do
  begin
    if gvcor^[j].warna<=0 then
    begin
      if gvcor^[j].warna>=-1e-4 then warna[j]:=1
      else warna[j]:=2;
    end
    else begin
      if gvcor^[j].warna<=1e-4 then warna[j]:=3
      else warna[j]:=4;
    end;
  end;
  printer.canvas.pen.color:=clwhite;
  printer.canvas.Brush.Color:=clblack;
  for i:=1 to gnum-1 do
  begin
    num:=(i-1)*gprt+1;
    num2:=(i)*gprt+1;
    polyvar[1].y:=round(sc[1]^[num]);
    polyvar[1].x:=round(sc[2]^[num]);
    polyvar[2].y:=round(sc[1]^[num2]);
    polyvar[2].x:=round(sc[2]^[num2]);
    for j:=2 to gprt do
    begin
```

```
      polyvar[4].y:=round(sc[1]^[num-1+j]);
      polyvar[4].x:=round(sc[2]^[num-1+j]);
      polyvar[3].y:=round(sc[1]^[num2-1+j]);
      polyvar[3].x:=round(sc[2]^[num2-1+j]);
      if (warna[num-2+j]<>warna[num-1+j])and(num-2+j>0)then
      begin
        if warna[num-2+j]=1 then printer.canvas.Brush.style:=bshorizontal;
        if warna[num-2+j]=2 then printer.canvas.Brush.style:=bsvertical;
        if warna[num-2+j]=3 then printer.canvas.Brush.style:=bsfdiagonal;
        if warna[num-2+j]=4 then printer.canvas.Brush.style:=bsbdiagonal;
      end;
      printer.canvas.polygon(slice(polyvar,4));
      polyvar[1].y:=polyvar[4].y;
      polyvar[1].x:=polyvar[4].x;
      polyvar[2].y:=polyvar[3].y;
      polyvar[2].x:=polyvar[3].x;
    end;
  end;
end;
for j:=1 to 2*bdprt do
onvert(mode,sscor[1]^[j],sscor[2]^[j],sscor[3]^[j],sc[1]^[j],sc[2]^[j]);
printer.canvas.pen.color:=clblack;
y1:=round(sc[1]^[1]);
x1:=round(sc[2]^[1]);
printer.canvas.moveto(x1,y1);
for j:=2 to bdprt do
begin
  y1:=round(sc[1]^[j]);
  x1:=round(sc[2]^[j]);
  printer.canvas.lineto (x1,y1);
end;
y1:=round(sc[1]^[1]);
x1:=round(sc[2]^[1]);
printer.canvas.moveto(x1,y1);
for j:=1 to bdprt do
begin
  y1:=round(sc[1]^[bdprt+j]);
  x1:=round(sc[2]^[bdprt+j]);
  printer.canvas.lineto (x1,y1);
end;

if showbd then
begin
  for j:=1 to bdnum*bdprt do
onvert(mode,bdcor[1]^[j],bdcor[2]^[j],bdcor[3]^[j],sc[1]^[j],sc[2]^[j]);
  if full then
    for j:=1 to bdnum*bdprt do
onvert(mode,bdcor[1]^[j],-bdcor[2]^[j],bdcor[3]^[j],sc2[1]^[j],sc2[2]^[j]);

  printer.canvas.pen.color:=clyellow;
  for i:=1 to bdnum do
  begin
    num:=(i-1)*bdprt+1;
    y1:=round(sc[1]^[num]);
    x1:=round(sc[2]^[num]);
    printer.canvas.  moveto(x1,y1);
    for j:=2 to bdprt do
    begin
      y1:=round(sc[1]^[num-1+j]);
       x1:=round(sc[2]^[num-1+j]);
      printer.canvas.lineto(x1,y1);
    end;
    if full then
    begin
      y1:=round(sc[1]^[num]);
```

```
      x1:=round(sc[2]^[num]);
      printer.canvas.moveto(x1,y1);
      for j:=1 to bdprt do
      begin
        y2:=round(sc2[1]^[num-1+j]);
        x2:=round(sc2[2]^[num-1+j]);
        printer.canvas.lineto (x2,y2);
      end;
    end;
  end;{for pol}
end;

printer.canvas.pen.color:=clred;
if showwl then
begin
  for j:=1 to wlprt*wlnum do
convert(mode,wlcor[1]^[j],wlcor[2]^[j],wlcor[3]^[j],sc[1]^[j],sc[2]^[j]);
  if full then
    for j:=1 to wlprt*wlnum do
convert(mode,wlcor[1]^[j],-wlcor[2]^[j],wlcor[3]^[j],sc2[1]^[j],sc2[2]^[j]);
  for i:=1 to wlnum do
  begin
    num:=(i-1)*wlprt+1;
    y1:=round(sc[1]^[num]);
    x1:=round(sc[2]^[num]);
    printer.canvas.moveto(x1,y1);
    for j:=2 to wlprt do
    begin
      y1:=round(sc[1]^[num-1+j]);
      x1:=round(sc[2]^[num-1+j]);
      printer.canvas.lineto(x1,y1);
    end;
    if full then
    begin
      y1:=round(sc[1]^[num]);
      x1:=round(sc[2]^[num]);
      printer.canvas.moveto(x1,y1);
      for j:=1 to wlprt do
      begin
        y2:=round(sc2[1]^[num-1+j]);
        x2:=round(sc2[2]^[num-1+j]);
        printer.canvas.lineto(x2,y2);
      end;
    end;
  end;{for nwl}
end;

if showbl then
begin
  for j:=1 to blprt*blnum do
convert(mode,blcor[1]^[j],blcor[2]^[j],blcor[3]^[j],sc[1]^[j],sc[2]^[j]);
  if full then
    for j:=1 to blprt*blnum do
convert(mode,blcor[1]^[j],-blcor[2]^[j],blcor[3]^[j],sc2[1]^[j],sc2[2]^[j]);

  printer.canvas.pen.color:=RGB(0,150,255);
  for i:=1 to blnum do
  begin
    num:=(i-1)*blprt+1;
    y1:=round(sc[1]^[num]);
    x1:=round(sc[2]^[num]);
    printer.canvas.moveto(x1,y1);
    for j:=2 to blprt do
    begin
      y1:=round(sc[1]^[num-1+j]);
```

```
      x1:=round(sc[2]^[num-1+j]);
      printer.canvas.lineto(x1,y1);
    end;
    if full then
    begin
      y1:=round(sc2[1]^[num]);
      x1:=round(sc2[2]^[num]);
      printer.canvas.moveto(x1,y1);
      for j:=1 to blprt do
      begin
        y2:=round(sc2[1]^[num-1+j]);
        x2:=round(sc2[2]^[num-1+j]);
        printer.canvas.lineto(x2,y2);
      end;
    end;
  end;
end;
d;


ocedure TFormview.Print1Click(Sender: TObject);
gin
printer.beginDoc;
printline(mode);
printer.Enddoc;
d;

ocedure TFormview.FormActivate(Sender: TObject);
gin
if not(pertama) then
bitblt(canvas.handle,5,50,780,540,
   bufimage.canvas.handle,0,0,srccopy);
d;

ocedure TFormview.Edit1Exit(Sender: TObject);
gin
if stringisinteger(edit1.text)then
begin
  if theta0<>strtofloat(edit1.text)then
  begin
    theta0:=strtofloat(edit1.text);
    tukarmode:=true;
  end;
end
else begin
   edit1.Text:=floattostr(theta0);
end;
d;

ocedure TFormview.Edit1KeyPress(Sender: TObject; var Key: Char);
gin
if key=#13 then
begin
  if stringisinteger(edit1.text)then
  begin
    theta0:=strtofloat(edit1.text);
    tukarmode:=true;
  end
  else begin
    edit1.Text:=floattostr(theta0);
  end;
end;
d;
```

```
cedure TFormview.Polygon1Click(Sender: TObject);
in
f not(Polygon1.checked) then
egin
  Polygon1.checked:=true;
  Viewpolyvert:=true;
nd
lse begin
  Polygon1.checked:=false;
  viewpolyvert:=false;
nd;
ukarmode:=true;
l;

cedure TFormview.Select1Click(Sender: TObject);
in
electcondition:=true;
l;

cedure TFormview.FormMouseDown(Sender: TObject; Button: TMouseButton;
hift: TShiftState; X, Y: Integer);

: x1,y1:array[1..4]of integer;
  i,j,num2,j1,i1,i2,i3,sign12,Upolbawah,Wpolbawah,Upolatas,Wpolatas:integer;
  xyinpoly,satisfiedeq:boolean;
  epsimin,epsi :integer;
in
if viewgausscurv then
if selectcondition then
egin
  for j:=1 to jlh_WL*jlh_sta do
nvert(mode,b[1]^[j],b[2]^[j],b[3]^[j],sc[1]^[j],sc[2]^[j]);
  for i:=1 to jlh_sta-1 do
  begin
    num:=(i-1)*jlh_wl+1;
    num2:=i*jlh_wl+1;
    x1[1]:=round(sc[1]^[num]);
    y1[1]:=round(sc[2]^[num]);
    x1[2]:=round(sc[1]^[num2]);
    y1[2]:=round(sc[2]^[num2]);
    for j:=2 to jlh_WL do
    begin
      x1[4]:=round(sc[1]^[num-1+j]);
      y1[4]:=round(sc[2]^[num-1+j]);
      x1[3]:=round(sc[1]^[num2-1+j]);
      y1[3]:=round(sc[2]^[num2-1+j]);
      xyinpoly:=true;
      for j1:=1 to 4 do
      begin
        i1:=j1;
        i2:=j1+1;if i2>4 then i2:=i2-4;
        i3:=j1+2;if i3>4 then i3:=i3-4;
        GetSign(x1[i1],y1[i1],x1[i2],y1[i2],x1[i3],y1[i3],sign12);
        Testsatisfied(x1[i1],y1[i1],x1[i2],y1[i2],x-5,y-50,sign12,satisfiedeq);
        if not(satisfiedeq) then
        begin
          xyinpoly:=false;
        end
      end;
      x1[1]:=x1[4];
      y1[1]:=y1[4];
      x1[2]:=x1[3];
      y1[2]:=y1[3];
      if xyinpoly then
      begin
```

```
        UpolNo:=i;WPolNO:=j-1;
      end;
    end;
  end;
  bufimage.canvas.pen.Color:=clred;
  bufimage.canvas.pen.Width:=2;
  Upolbawah:=UpolNo-1;Upolatas:=UpolNo+2;
  Wpolbawah:=WpolNo-1;Wpolatas:=WpolNo+2;
  if UpolBawah<1 then Upolbawah:=1;
  if Upolatas>jlh_sta then Upolatas:=jlh_sta;
  if wpolBawah<1 then wpolbawah:=1;
  if wpolatas>jlh_WL then wpolatas:=jlh_WL;
  for i:=Upolbawah to Upolatas-1 do
  begin
    for j:=WPolbawah to Wpolatas-1 do
    begin
      num:=(i-1)*jlh_wl+j;
      num2:=i*jlh_wl+j;
      x1[1]:=round(sc[1]^[num]);
      y1[1]:=round(sc[2]^[num]);
      x1[2]:=round(sc[1]^[num2]);
      y1[2]:=round(sc[2]^[num2]);
      x1[4]:=round(sc[1]^[num+1]);
      y1[4]:=round(sc[2]^[num+1]);
      x1[3]:=round(sc[1]^[num2+1]);
      y1[3]:=round(sc[2]^[num2+1]);
      bufimage.canvas.moveto(x1[1],y1[1]);
      bufimage.canvas.lineto(x1[2],y1[2]);
      bufimage.canvas.lineto(x1[3],y1[3]);
      bufimage.canvas.lineto(x1[4],y1[4]);
      bufimage.canvas.lineto(x1[1],y1[1]);
    end;
  end;
end;
selectcondition:=false;
if changepolystatus then
begin
  for j:=1 to jlh_WL*jlh_sta do
onvert(mode,b[1]^[j],b[2]^[j],b[3]^[j],sc[1]^[j],sc[2]^[j]);
  epsimin:=10;changepolyada:=false;
  for i:=Upolno-1 to Upolno+2 do
  begin
    for j:=1 to 4 do
    begin
      num:=(i-1)*jlh_wl+WPolNo-2+j;
      epsi:=abs(x-5-round(sc[1]^[num]))+abs(y-50-round(sc[2]^[num]));
      if epsi<epsimin then
      begin
        changepolyada:=true;
        epsimin:=epsi;NoPolychange:=num;
      end;
    end;
  end;
  if changepolyada then
  begin

ufimage.canvas.rectangle(round(sc[1]^[Nopolychange])-2,round(sc[2]^[Nopolychange])-2,
  round(sc[1]^[Nopolychange])+2,round(sc[2]^[Nopolychange])+2);
  with formCP do
  begin
    edit1.text:=formatfloat('0.0000',b[1]^[Nopolychange]);
    edit2.text:=formatfloat('0.0000',b[2]^[Nopolychange]);
    bawal:=b[2]^[NOPOlychange];
    edit3.text:=formatfloat('0.0000',b[3]^[Nopolychange]);
    Onpolychange;
```

```
  end;
  end;
nd;
itblt(canvas.handle,5,50,780,540,
   bufimage.canvas.handle,0,0,srccopy);
hangepolystatus:=false;
i;

cedure TFormview.ChangePolygon1Click(Sender: TObject);
in
hangepolystatus:=true;
ormCP.show;
i;

cedure TFormview.Edit2Exit(Sender: TObject);
in
if stringisinteger(edit2.text)then
egin
  if tilt<>strtofloat(edit2.text)then
  begin
    tilt:=strtofloat(edit2.text);
    phi0:=tilt+90;
    tukarmode:=true;
  end;
nd
lse begin
   edit2.Text:=floattostr(tilt);
nd;
d;

ocedure TFormview.Edit2KeyPress(Sender: TObject; var Key: Char);
in
if key=#13 then
egin
  if stringisinteger(edit2.text)then
  begin
    tilt:=strtofloat(edit2.text);
    phi0:=tilt+90;
    tukarmode:=true;
  end
  else begin
    edit2.Text:=floattostr(tilt);
  end;
nd;
d;

d.
```